MGS613 Group 11 Final Project

Aryansh Fooria Blessen John Gongle Tejaswini Hiren Avera Mansi Nagar Simran Pahuja

"It's not about ideas. It's about making ideas happen." — Scott Belsky

Introduction about Olist

Olist, a Brazilian startup started its operations in 2015 with a clear goal to help small businesses grow by connecting them with large online marketplaces. Founder Tiago Dalvi's idea behind Olist was to give smaller stores access to a broader customer base so they could compete with big retailers, through a subscription platform Software-as-a-Service (SaaS). Thereby Olist helped small stores manage their products, inventory, and logistics in a more organized way, which allowed these businesses to reach more customers without needing their own e-commerce infrastructure.

Olist easy and user-friendly interface helped to grow strongly gaining thousands of new users. By 2019, Olist had over 7,000 small businesses were using its platform to sell directly to customers. The Covid-19 pandemic led to wider acceptance of e-commerce by both sellers and buys, leading more people to shop online frequently. These factors helped Olist's revenue to soar to new highs, with sales figures tripling from previous year in just the first half of 2021.

Olist's main product 'Olist Store', gave merchants tools to list products, handle payments, and manage shipping. By partnering with popular Latin American platforms like Mercado Livre and B2W, Olist provided sellers with a huge reach across the region. Olist's success inspired the launch of another tool, 'Olist Shops', in 2020. This product enabled anyone to create an online store in just a few minutes, complete with payment and delivery options and attracted over 200,000 users globally within a year.

By 2021, Olist's rapid growth story caught the attention of major global investors. The company was able to raise $186 million in funding, achieving a "Unicorn" status (a valuation of over $1 billion). This funding, backed by investors like SoftBank and Goldman Sachs, helped Olist expand its services, especially in logistics, by acquiring companies like Pax Logistica and Clickspace. This move helped Olist build a solid delivery network with over 4,000 drivers.

Olist went public in May 2024, a major achievement that highlighted its market position and progress in just nine years of operation. With online shopping becoming increasingly popular and a new normal, Olist continues to give small Brazilian businesses the tools they need to grow and compete on a larger stage.

Project Goals

The project aims to analyze the sales volume and trends of Olist Store in the city of Rio de Janeiro, Brazil. It seeks to identify the top-selling product mix, determine the average customer spend on the platform, assess preferred payment methods, and evaluate average customer satisfaction based on reviews given.

The insights derived from this analysis will help Olist make informed decisions to enhance sales performance, reward sellers with high sales volumes and customer satisfaction, and optimize their payment gateway based on customer preferences and utilization for a seamless shopping experience.

Business Model

Olist connects small businesses to sell their products across Brazil without any hassle. When a customer buys something from the Olist Store, the seller gets a notification to process the order

and ships the product directly to the customer. After the customer receives their purchase (or when the expected delivery date arrives), they get an email survey. This survey allows them to rate their shopping experience and leave feedback, which helps Olist and its sellers improve the service and product quality.

About the Data

We obtained the Olist dataset from Kaggle, which contains a total of 100,000 order records spanning their limited operating period from 2016 to 2018. For the purposes of our project, we will focus on a subset of the data. The dataset provides comprehensive information on sellers, customers, orders, products, order items, payments, and performance metrics, such as customer locations and reviews. It also includes geolocation details that map Brazilian zip codes to latitude and longitude coordinates.

Due to data importing and processing limitations in the free version of Oracle Apex, we filtered the dataset to include only records from Rio de Janeiro. This decision was based on Rio de Janeiro's status as the second-largest city in Brazil, both in terms of population and Gross Domestic Product, after São Paulo. Its economic significance makes it a high-priority market for Olist's growth and development.

Business Rules

1. Each order is associated with only one seller.
2. Each seller has one or multiple orders.
3. Each order would be associated to a new customer.
4. Customers can be linked to a Customer Unique ID for aggregation. A Customer Unique ID can have multiple orders.
5. Each order contains multiple items, but each item in an order must correspond to a specific product within that order.
6. Each seller has one or more than one product item.
7. There can be one or more than one product item associated with a single seller.
8. Each order can contain multiple instances/items of a product, and each product in an order must be represented as a separate order item.
9. Each order must be combined with one or more payment sequential related to one payment type.
10. Each payment must be linked to an order.
11. Each order can be "Delivered" or "Not Delivered".
12. Each delivery status is related one and only one order.
13. Regardless of order status, payment would always be made by the customer. However, the product may or may not delivered even if the payment is made.
14. One delivery status would be attached to one and only one product, but the product may or may not be delivered to the customer.
15. Each "Delivered" order must have a review score.
16. Review Score present only if the order is delivered.

Reference Materials

Link to the source data from Kaggle:
https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=olist_products_dataset.csv

Link to Google Drive with data sets used in Oracle Apex
https://drive.google.com/drive/folders/1WG2j4YJIBEiwJdT_YYfa5fxOdTsk83cY?usp=sharing

Data Dictionary

Attribute Name	Data Type	Description
customer_id	Varchar (32)	A Unique ID for each order
customer_unique_id	Varchar (32)	A Unique ID for each customer
customer_zip_code_prefix	Char (5)	The first five digits of the customer's ZIP code
customer_city	Varchar (50)	The city where the customer is located.
customer_state	Char (2)	The state where the customer is located.
geolocation_zip_code_prefix	Char (5)	The first five digits of the ZIP code for a geographic location
geolocation_lat	Float	The latitude of a location
geolocation_lng	Float	The longitude of a location
geolocation_city	Varchar (50)	The city associated with the geolocation ZIP code,
geolocation_state	Char (2)	The state associated with the geolocation ZIP code.
order_item_id	Integer	A Unique ID for each item within an order
shipping_limit_date	Date	The latest date by which the seller is required to ship the item to meet the expected delivery timeline.
price	Float	The price of the individual product within the order, before any shipping costs.
freight_value	Float	The cost of shipping for the item
payment_sequential	Integer	The sequence number of the payment for a specific order
payment_type	Varchar (20)	The method of payment used for the transaction
payment_installments	Integer	The number of instalments in which the payment is divided
payment_value	Float	The total amount paid for the order
review_id	Varchar (32)	A Unique ID for each review submitted by customers
review_score	Integer	A numerical rating given by the customer
review_creation_date	Date	The date when the review was submitted
review_answer_timestamp	Date	The timestamp indicating when a response to the review was provided by the seller or the customer service team.
order_id	Varchar (32)	A Unique ID for each customer placing an order.
order_status	Varchar (20)	The current status of the order
order_purchase_timestamp	Datetime	The date and time when the order was placed.
order_approved_at	Datetime	The date and time when the order was approved for processing.
order_delivered_carrier_date	Date	The date when the order was handed over to the delivery carrier.
order_delivered_customer_date	Date	The date when the customer actually received the order.
order_estimated_delivery_date	Date	The estimated date when the order is expected to be delivered to the customer.
product_id	Varchar (32)	A Unique ID for each product in the catalog.
product_category_name	Varchar (50)	The unique name of the category to which the product belongs in Portuguese
product_name_lenght	Integer	The length of the product name, measured in characters.
product_description_lenght	Integer	The length of the product description, measured in characters
product_photos_qty	Integer	The number of photos available for the product.
product_weight_g	Integer	The weight of the product in grams.
product_length_cm	Integer	The length of the product in centimeters.
product_height_cm	Integer	The height of the product in centimeters.
product_width_cm	Integer	The width of the product in centimeters.
seller_id	Varchar (32)	A Unique ID for each seller in the marketplace.
seller_zip_code_prefix	Char (5)	The first five digits of the seller's zip code
seller_city	Varchar (50)	The city where the seller is located.
seller_state	Char (2)	The state where the seller operates.

Table 1: Data Dictionary

Entity Relationship Diagram

At first glance, the Entity Relationship (ER) Diagram may appear to have several entities that can be merged. However, the structure of the Olist dataset provided and the nature of its data values limit the feasibility of merging entities. The way we have designed the ER diagram, minimizes data duplication errors and optimizes the data processing sequence, ensuring efficiency and accuracy in analysis.

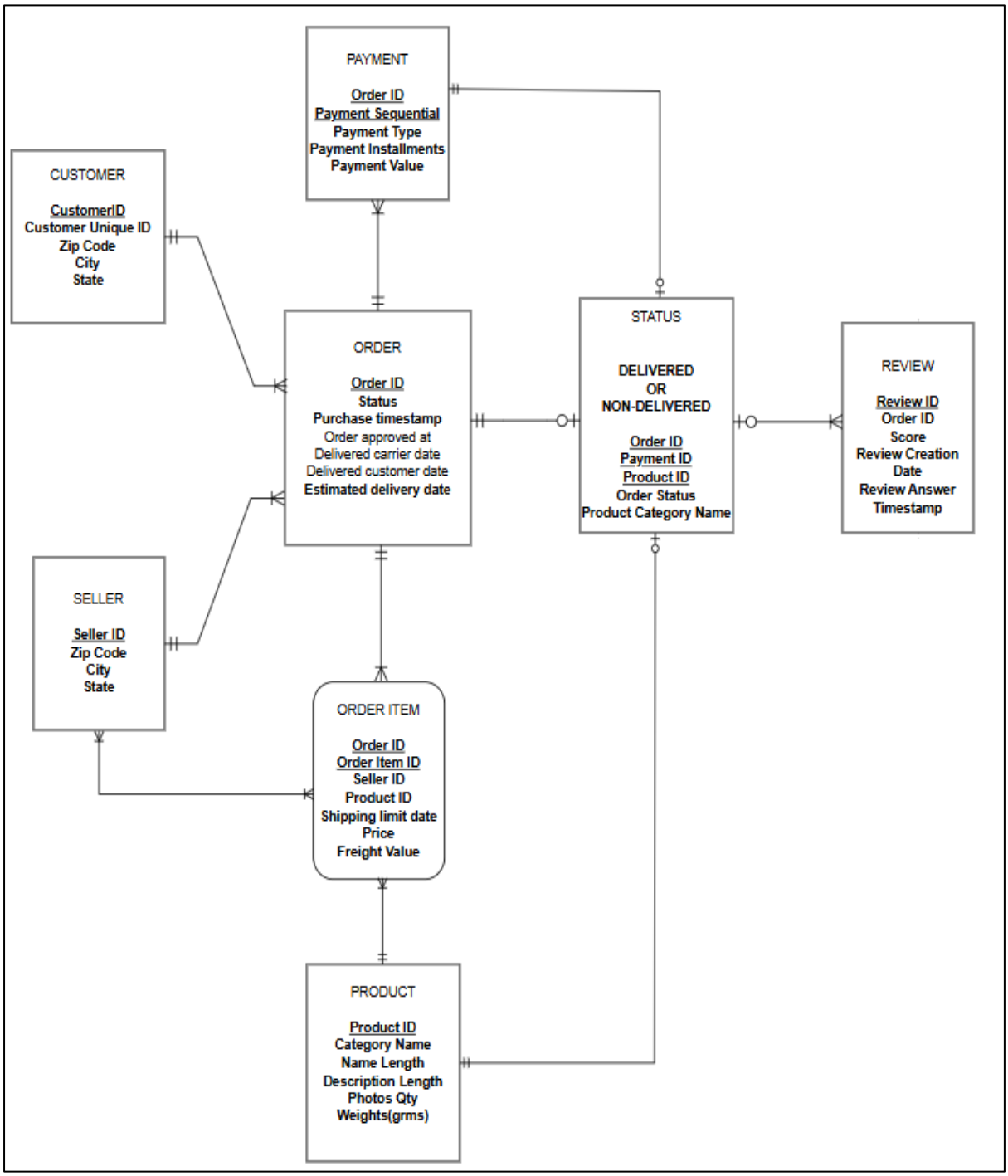


Figure 1: ER Diagram

Relational Schema

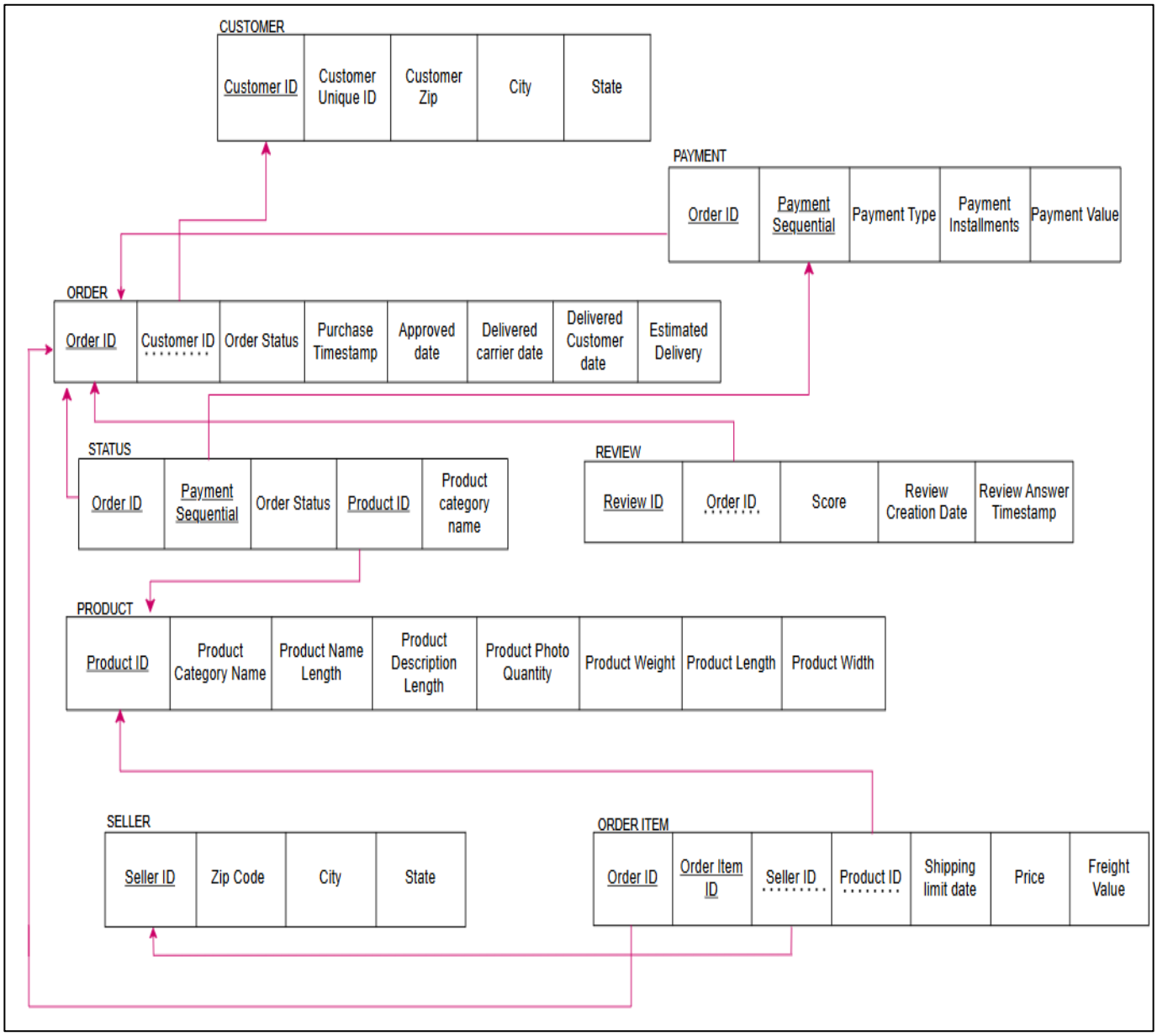


Figure 2: R-Schema

Data Cleaning

The Olist dataset has very few null values, and most of the columns are meaningful and valuable. However, while examining the Olist dataset, we found majorly empty (null) values in the "review_comment_title" and "review_comment_message" columns. Since these comments were written in Portuguese, we decided to drop them, as we can rely on the "review score" column alone to assess the company's delivery and product quality.

The products table contained a significant number of null values across seven columns, including "product_category_name, product_name_length, product_description_length, product_photos_qty, product_weight_g, product_length_cm, and product_width_cm". To address the null values, we replaced the empty cells in product_category_name with 'N/A' and filled the other six columns with '0', ensuring uniform data types across the columns.

Data Application and Utilization

For building Entity Relationship Diagram, we have utilized six strong entities namely SELLER, ORDER, CUSTOMER, PRODUCT, PAYMENT, REVIEW and STATUS. The Primary Key for these tables is *seller id, order id, customer id, product id, review id, payment id,* (created based on composite keys of order id and payment sequential)*, Status* (created based on composite keys of order id, product id and payment id). As there is a many-to-many relationship between the ORDER and PRODUCT entities, we have introduced an associative entity, ORDER ITEM, for normalization and a proper data structure.

Data Count and Justification

Due to data import and processing constraints on the free student version of Oracle Apex, we filtered the Customer ID dataset to include only customers based in Rio de Janeiro, which is the second-most-populous city located in Brazil. This resulted in 6,882 customer IDs, representing 6,620 unique customers from this location on the platform.

Using this filtered customer data, we extracted the corresponding 6,882 order IDs from the Order dataset and further filtered related data from other tables to focus exclusively on Rio de Janeiro.

Table Creation and Data Importing

After creating the tables in Oracle Apex using the SQL statements provided below, we imported the data directly using the 'Load Data' function. Given the large volume of data, the 'Insert Data' function was not a practical option.

Entity	Create Table Statements for the Entity
Customer	Create table OLIST_CUSTOMER (customer_id VARCHAR2(100) NOT NULL, customer_unique_id VARCHAR2(100) NOT NULL, customer_zip_code_prefix CHAR(5) NOT NULL, customer_city VARCHAR2(50) NOT NULL, customer_state CHAR(2) NOT NULL, CONSTRAINT OLIST_CUST PRIMARY KEY (customer_id));
Order	Create table OLIST_ORDER(customer_id VARCHAR2(100) NOT NULL, order_id VARCHAR2(100) NOT NULL, order_status VARCHAR2(50) NOT NULL, order_purchase_timestamp TIMESTAMP, order_approved_at TIMESTAMP, order_delivered_carrier_date TIMESTAMP, order_delivered_customer_date TIMESTAMP, order_estimated_delivery_date TIMESTAMP NOT NULL, CONSTRAINT OLIST_ORDER_PK_RDJ PRIMARY KEY (order_id), CONSTRAINT OLIST_ORDER_FK_RDJ FOREIGN KEY (customer_id) REFERENCES OLIST_CUSTOMER(customer_id));
Seller	Create table OLIST_SELLER(seller_id VARCHAR2(100) NOT NULL, seller_zip_code_prefix CHAR(5) NOT NULL, seller_city VARCHAR2(50) NOT NULL, seller_state CHAR(2) NOT NULL, CONSTRAINT OLIST_SELLER_PK PRIMARY KEY (seller_id));

Product	Create table OLIST_PRODUCT(product_id VARCHAR2(100) NOT NULL, product_category_name VARCHAR2(100), product_name_lenght CHAR(2), product_description_lenght VARCHAR2(5), product_photos_qty VARCHAR2(2), product_weight_g VARCHAR2(10), product_length_cm VARCHAR2(10), product_height_cm VARCHAR2(10), product_width_cm VARCHAR2(10), CONSTRAINT OLIST_PRODUCT_PK PRIMARY KEY (product_id));
Payment	Create table OLIST_PAYMENT(order_id VARCHAR2(100) NOT NULL, payment_sequential VARCHAR2(5) NOT NULL, payment_type VARCHAR2(100) NOT NULL, payment_installments VARCHAR2(5) NOT NULL, payment_value NUMBER(*,2), CONSTRAINT OLIST_ORDER_PAYMENT_PK PRIMARY KEY (order_id,payment_sequential), CONSTRAINT OLIST_ORDER_PAYMENT_FK FOREIGN KEY (order_id) REFERENCES OLIST_ORDER(order_id));
Order Item	Create table OLIST_ORDER_ITEM(order_id VARCHAR2(100) NOT NULL, order_item_id VARCHAR2(10) NOT NULL, product_id VARCHAR2(100) NOT NULL, seller_id VARCHAR2(100) NOT NULL, shipping_limit_date DATE NOT NULL , price NUMBER(*,2), freight_value NUMBER(*,2), CONSTRAINT OLIST_ORDER_ITEM_PK PRIMARY KEY (order_id,order_item_id), CONSTRAINT OLIST_ORDER_ITEM_FK FOREIGN KEY (order_id) REFERENCES OLIST_ORDER(order_id));
Status	Create table OLIST_STATUS as Select distinct o.order_id, p.payment_sequential, o.order_status, oi.product_id, pr.product_category_name from OLIST_ORDER o left join OLIST_PAYMENT p on o.order_id = p.order_id join olist_order_item oi on o.order_id = oi.order_id join olist_product pr on oi.product_id = pr.product_id;
Review	Create table OLIST_REVIEW(review_id VARCHAR2(100) NOT NULL, order_id VARCHAR2(100) NOT NULL, review_score CHAR(5) NOT NULL, review_comment_title VARCHAR2(50), review_comment_message VARCHAR2(300), review_creation_date TIMESTAMP NOT NULL, review_answer_timestamp TIMESTAMP NOT NULL, CONSTRAINT OLIST_REVIEW_PK PRIMARY KEY (review_id), CONSTRAINT OLIST_REVIEW_FK FOREIGN KEY (order_id) REFERENCES OLIST_ORDER(order_id));

Analysis on Olist - Rio De Janeiro

Order Related:

1. How many orders were placed on Olist from the state of Rio De Janerio, Brazil and how does this vary by month or season?

Total Orders:

```
SELECT
    COUNT(order_id) AS TOTAL_ORDERS
FROM
    OLIST_ORDER;
```

TOTAL_NUMBER_OD_ORDERS
6882

Extracting Total Order Month over Month:

```
SELECT
   EXTRACT(YEAR FROM order_purchase_timestamp) AS Order_Year,
   EXTRACT(MONTH FROM order_purchase_timestamp) AS Order_Month,
   COUNT(order_id) AS Total_Orders
FROM
   OLIST_ORDER
GROUP BY
   EXTRACT(YEAR FROM order_purchase_timestamp),
   EXTRACT(MONTH FROM order_purchase_timestamp)
ORDER BY
   Order_Year,
   Order_Month ASC;
```

ORDER_YEAR	ORDER_MONTH	TOTAL_ORDERS
2016	11	26
2016	12	12
2017	2	23
2017	3	218
2017	4	172
2017	5	212
2017	6	267
2017	7	226
2017	8	348
2017	9	329
2017	10	308
2017	11	352
2017	12	578
2018	1	323
2018	2	474
2018	3	528
2018	4	370
2018	5	365
2018	6	460
2018	7	328
2018	8	800
2018	9	159
2018	10	4

The order data available on Kaggle covers a limited time period, making it difficult to conduct a clear trend analysis for year-over-year comparisons, especially for a company in its growth stage. However, the data does indicate strong month-over-month improvements in sales up until August 2018. Notably, Olist faced performance issues in Q3 2018 due to late deliveries. Additionally, the data for October 2018 includes observations only up to the 5th of the month, resulting in artificially low figures for that period.

2. What is the total order that is either not delivered or cancelled on Olist?

```
SELECT
   ORDER_STATUS,
   COUNT(ORDER_ID) AS Total_Orders,
   ROUND(COUNT(ORDER_ID) / (SELECT COUNT(ORDER_ID)
                  FROM OLIST_ORDER) * 100, 2) || '%' AS Average_Not_Delivered_Orders
FROM
   OLIST_ORDER
WHERE
   ORDER_STATUS NOT IN ('delivered')
GROUP BY
   ORDER_STATUS;
```

ORDER_STATUS	TOTAL_ORDERS	AVERAGE_NOT_DELIVERED_ORDERS
shipped	170	2.47%
processing	15	.22%
invoiced	14	.2%
unavailable	33	.48%
created	1	.01%
canceled	48	.7%

Olist has demonstrated strong performance, with a cancellation rate of just 0.7% for the entire reviewed time period. This is significantly lower than the average cancellation rate for online retailers, which typically ranges from 2% to 5%.

From the analysis of the entire order history, we observed that only 4.08% of orders were not delivered, with the majority (2.47%) being orders that were shipped but not completed. This indicates an opportunity for Olist to revamp its operational logistics to improve delivery times. However, it is important to note that transportation and logistics in Brazil present significant challenges due to difficult terrain and limited infrastructure.

Product and Sales Related:

1. What is the top-selling product categories on Olist, both in terms of sales revenue and product quantity?

Top 5 Product Categories by Sales Revenue

```
SELECT
   PRODUCT_CATEGORY_NAME,
   ROUND(SUM(PRICE), 0) AS Total_Revenue
FROM
   OLIST_PRODUCT A
LEFT JOIN
   OLIST_ORDER_ITEM B
ON
   A.PRODUCT_ID = B.PRODUCT_ID
GROUP BY
   PRODUCT_CATEGORY_NAME
ORDER BY
   Total_Revenue DESC
FETCH FIRST 5 ROWS ONLY;
```

PRODUCT_CATEGORY_NAME	TOTAL_REVENUE
relogios_presentes	93029
beleza_saude	86558
cama_mesa_banho	77871
esporte_lazer	70763
informatica_acessorios	64797

Gift Watches emerged as the top-selling item by revenue. Other top product categories include Beauty & Health, Bed Accessories, Leisure & Sports, and Computer Accessories

Top 5 Product Category by Sales Quantity

```
SELECT
   PRODUCT_CATEGORY_NAME,
   COUNT(ORDER_ID) AS Total_Count
FROM
   OLIST_PRODUCT A
LEFT JOIN
   OLIST_ORDER_ITEM B
ON
   A.PRODUCT_ID = B.PRODUCT_ID
GROUP BY
   PRODUCT_CATEGORY_NAME
ORDER BY
   Total_Count DESC
FETCH FIRST 5 ROWS ONLY;
```

PRODUCT_CATEGORY_NAME	TOTAL_COUNT
cama_mesa_banho	854
moveis_decoracao	615
esporte_lazer	593
beleza_saude	572
informatica_acessorios	538

Bed Accessories sold the highest quantity, followed by Furniture Decorations, indicating a strong presence of home décor sellers on the platform. Other top product categories by quantity include Leisure & Sports, Beauty & Health, and Computer Accessories.

Notably, Bed Accessories, Leisure & Sports, Beauty & Health, and Computer Accessories are leading product categories in terms of both revenue and quantity.

2. What is the average order value (AOV) on Olist, and how does this vary by product category or payment method?

Average Order Value

```
SELECT
   ROUND((SUM(PRICE) / COUNT(ORDER_ID)), 0) AS AOV
FROM
   OLIST_ORDER_ITEM;
```

AOV
127

The average order value on the platform is 127 Brazilian Reais (BRL), which is significantly lower than the average in Brazil's general e-commerce market. To improve this, Olist should focus on offering higher-value products and explore exclusive deals with partners to enhance its product offerings and increase the average order value. A particular focus on electronics could be a good starting point, given the higher average order value in this category.

Product Category Wise

```
SELECT
   PRODUCT_CATEGORY_NAME,
   ROUND((SUM(PRICE) / COUNT(ORDER_ID)), 0) AS AOV
FROM
   OLIST_PRODUCT A
LEFT JOIN
   OLIST_ORDER_ITEM B
ON
   A.PRODUCT_ID = B.PRODUCT_ID
GROUP BY
   PRODUCT_CATEGORY_NAME
ORDER BY
   AOV DESC
FETCH FIRST 5 ROWS ONLY;
```

PRODUCT_CATEGORY_NAME	AOV
pcs	874
eletrodomesticos_2	768
telefonia_fixa	686
portateis_casa_forno_e_cafe	519
instrumentos_musicais	298

Among product categories, Computers and Laptops have the highest average order value at 874 BRL, followed by other electronic items such as Home Appliances, Phones, and Coffee Machines. Musical Instruments is the only non-electronic item to feature on this list.

Payment Type Based

```
SELECT
   PAYMENT_TYPE,
   ROUND((SUM(PRICE) / COUNT(B.ORDER_ID)), 0) AS AOV
FROM
   OLIST_PAYMENT A
LEFT JOIN
   OLIST_ORDER_ITEM B
ON
   A.ORDER_ID = B.ORDER_ID
GROUP BY
   PAYMENT_TYPE
ORDER BY
   AOV DESC;
```

PAYMENT_TYPE	AOV
credit_card	132
debit_card	114
boleto	111
voucher	91

Orders placed using Credit Cards have the highest average order value at 132 BRL, followed by Debit Cards. Therefore, payment gateways focusing on credit card transactions should prioritize convenience and security to enhance the customer experience.

Order and Review Related:

1. Number and Percentage of Orders on Olist with a review score 5 and payment type as credit card?

Number and Percentage of 5 Rated Review Orders

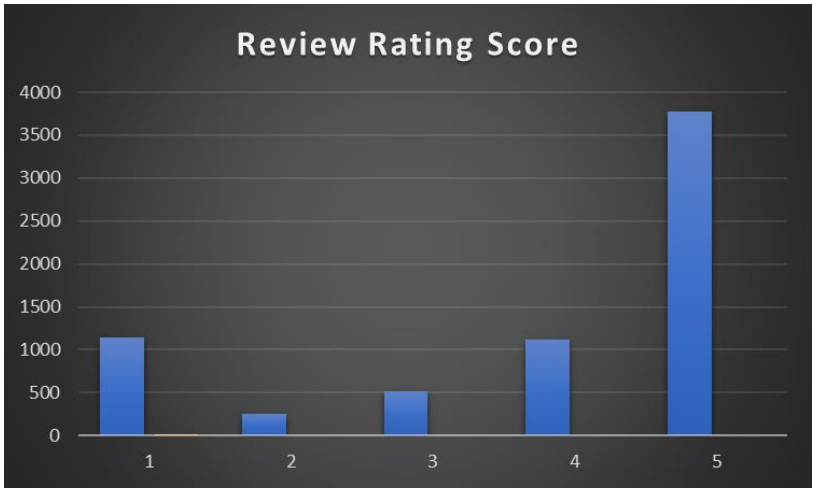
```
SELECT
   REVIEW_SCORE,
   COUNT(ORDER_ID) AS Number_Of_Orders,
   ROUND(COUNT(ORDER_ID) / (SELECT COUNT(ORDER_ID)
                  FROM OLIST_ORDER) * 100) || '%' AS Percentage_Order
FROM
   OLIST_REVIEW
WHERE
   REVIEW_SCORE = 5
GROUP BY
   REVIEW_SCORE;
```

PAYMENT_TYPE	NUMBER_OF_ORDERS	PERCENTAGE_ORDER
credit_card	5512	80%

Distribution of Rating and Percentage for All Orders

```
SELECT
   REVIEW_SCORE,
   COUNT(ORDER_ID) AS Number_Of_Orders,
   ROUND(COUNT(ORDER_ID) / (SELECT COUNT(ORDER_ID)
                  FROM OLIST_ORDER) * 100, 2) || '%' AS Percentage_Of_Orders
FROM
   OLIST_REVIEW
GROUP BY
   REVIEW_SCORE
ORDER BY
   REVIEW_SCORE;
```

REVIEW_SCORE	NUMBER_OF_ORDERS	PERCENTAGE_OF_ORDERS
1	1139	16.55%
2	250	3.63%
3	511	7.43%
4	1124	16.33%
5	3780	54.93%



Of the total orders placed, 55% have received a 5-star rating. While this is a strong performance, Olist should aim to improve this further and strive for an average rating above 4.5. It is crucial for Olist to address the underlying causes of 1-star ratings and take corrective actions to ensure continued success in the long term. Currently, the average rating for Rio de Janeiro is below 4, standing at 3.9.

Number of 5 Rated Orders Placed Using Credit Card

```
SELECT
   PAYMENT_TYPE,
   COUNT(DISTINCT ORDER_ID) AS Number_Of_Orders,
   ROUND(COUNT(DISTINCT ORDER_ID) / (SELECT COUNT(DISTINCT ORDER_ID)
                        FROM OLIST_ORDER) * 100) || '%' AS Percentage_Order
FROM
   OLIST_PAYMENT
WHERE
   PAYMENT_TYPE = 'credit_card'
GROUP BY
   PAYMENT_TYPE;
```

PAYMENT_TYPE	NUMBER_OF_ORDERS	PERCENTAGE_ORDER
credit_card	5512	80%

80% of 5-star reviews were given by customers who used credit cards for payment. This suggests that the credit card payment option provides the most ease and convenience for customers when placing orders on the platform.

Seller Related

1. Identify Top 5 Sellers for the entire and observe their Quarter-on-Quarter Trend.

```
WITH TOP_SELLERS AS (
   SELECT SELLER_ID
   FROM OLIST_ORDER_ITEM
   GROUP BY SELLER_ID
   ORDER BY COUNT(DISTINCT ORDER_ID) DESC
   FETCH FIRST 5 ROWS ONLY
)
SELECT
   A.SELLER_ID,
   EXTRACT(YEAR FROM B.order_purchase_timestamp) AS Year,
   CASE
      WHEN EXTRACT(MONTH FROM B.order_purchase_timestamp) IN (1,2,3) THEN 'Quarter1'
      WHEN EXTRACT(MONTH FROM B.order_purchase_timestamp) IN (4,5,6) THEN 'Quarter2'
      WHEN EXTRACT(MONTH FROM B.order_purchase_timestamp) IN (7,8,9) THEN 'Quarter3'
      ELSE 'Quarter4'
   END AS Quarter,
   COUNT(DISTINCT A.ORDER_ID) AS NUMBER_OF_ORDERS
FROM
   OLIST_ORDER_ITEM A
LEFT JOIN
   OLIST_ORDER B
   ON A.ORDER_ID = B.ORDER_ID
WHERE
   A.SELLER_ID IN (SELECT SELLER_ID FROM TOP_SELLERS)
   AND EXTRACT(YEAR FROM B.order_purchase_timestamp) IN (2017, 2018)
GROUP BY
   A.SELLER_ID,
   EXTRACT(YEAR FROM B.order_purchase_timestamp),
   CASE
      WHEN EXTRACT(MONTH FROM B.order_purchase_timestamp) IN (1,2,3) THEN 'Quarter1'
      WHEN EXTRACT(MONTH FROM B.order_purchase_timestamp) IN (4,5,6) THEN 'Quarter2'
      WHEN EXTRACT(MONTH FROM B.order_purchase_timestamp) IN (7,8,9) THEN 'Quarter3'
      ELSE 'Quarter4'
   END
ORDER BY
   Year ASC,
   Quarter ASC,
   NUMBER_OF_ORDERS DESC;
```

SELLER_ID	YEAR	QUARTER	NUMBER_OF_ORDERS
7c67e1448b00f6e969d365cea6b010ab	2017	Quarter1	4
4a3ca9315b744ce9f8e9374361493884	2017	Quarter1	3
4a3ca9315b744ce9f8e9374361493884	2017	Quarter2	13
6560211a19b47992c3666cc44a7e94c0	2017	Quarter2	8
7c67e1448b00f6e969d365cea6b010ab	2017	Quarter2	7
1f50f920176fa81dab994f9023523100	2017	Quarter2	2
4869f7a5dfa277a7dca6462dcf3b52b2	2017	Quarter2	1
4a3ca9315b744ce9f8e9374361493884	2017	Quarter3	24
1f50f920176fa81dab994f9023523100	2017	Quarter3	23
7c67e1448b00f6e969d365cea6b010ab	2017	Quarter3	18
6560211a19b47992c3666cc44a7e94c0	2017	Quarter3	14
4869f7a5dfa277a7dca6462dcf3b52b2	2017	Quarter3	13
1f50f920176fa81dab994f9023523100	2017	Quarter4	43
4a3ca9315b744ce9f8e9374361493884	2017	Quarter4	35
4869f7a5dfa277a7dca6462dcf3b52b2	2017	Quarter4	25
7c67e1448b00f6e969d365cea6b010ab	2017	Quarter4	20
6560211a19b47992c3666cc44a7e94c0	2017	Quarter4	20
1f50f920176fa81dab994f9023523100	2018	Quarter1	28
4a3ca9315b744ce9f8e9374361493884	2018	Quarter1	25
4869f7a5dfa277a7dca6462dcf3b52b2	2018	Quarter1	21
7c67e1448b00f6e969d365cea6b010ab	2018	Quarter1	18
6560211a19b47992c3666cc44a7e94c0	2018	Quarter1	15
4869f7a5dfa277a7dca6462dcf3b52b2	2018	Quarter2	31
7c67e1448b00f6e969d365cea6b010ab	2018	Quarter2	25
1f50f920176fa81dab994f9023523100	2018	Quarter2	19
6560211a19b47992c3666cc44a7e94c0	2018	Quarter2	18
4a3ca9315b744ce9f8e9374361493884	2018	Quarter2	17
6560211a19b47992c3666cc44a7e94c0	2018	Quarter3	39
4a3ca9315b744ce9f8e9374361493884	2018	Quarter3	22
4869f7a5dfa277a7dca6462dcf3b52b2	2018	Quarter3	17
7c67e1448b00f6e969d365cea6b010ab	2018	Quarter3	16
1f50f920176fa81dab994f9023523100	2018	Quarter3	8

Upon reviewing the data, we observed that three of the top five sellers were listed on the platform only after Q1 2017. In 2017, Seller ID 4a3ca9315b744ce9f8e9374361493884 emerged as the top seller. However, in 2018 (for the three quarters analyzed), Seller ID 6560211a19b47992c3666cc44a7e94c0 took the lead as the top seller. Despite this, our analysis reveals that Seller ID 4a3ca9315b744ce9f8e9374361493884 has been the top-performing seller over the entire period analyzed.

With this data, Olist can recognize and partner with their top sellers, leveraging their success to attract new sellers through targeted marketing and incentivized sales. This approach can drive increased revenue generation in the long run.

Websites Referenced

- https://valorinternational.globo.com/business/news/2021/12/15/olist-is-brazils-newest-unicorn.ghtml
- https://techcrunch.com/2021/04/15/goldman-sachs-leads-23m-in-funding-for-brazilian-e-commerce-startup-olist/