Mahmoud Amin

OLIST
SQL EDA

https://github.com/mahmoudamintaha/olist_eda

INTRODUCTION

Here is the follow-up to the OLIST database project. In part 1, we gathered CSV files, cleaned tables, and established primary and foreign keys to build the desired schema. Now, we will delve into an exploratory data analysis using SQL to gain insights into customers, sellers, and states.

Table of CONTENTS

01

Customers Analysis

- What is the number of our registered customers?
- Which state has the highest number of registered customers?
- What are the top 5 cities based on registered customers?
- What are the 5 states with the least number of registered customers?
- Who are our top 100 customers based on total sales?
- What are the IDs of customers that didn't make any orders since registration?

WHICH STATE HAS THE HIGHEST NUMBER OF REGISTERED CUSTOMERS?

The result of this query Is Sao Paolo
Which makes sense since it’s the largest state in Brazil by population

```
%%sql
SELECT
  customer_state AS State,
  COUNT(DISTINCT customer_unique_id) Cust_num
FROM
  customers
GROUP BY
  1
ORDER BY
  2 DESC
LIMIT
  1
```

WHAT ARE THE 5 STATES WITH THE LEAST NUMBER OF REGISTERED CUSTOMERS?

The result of this query Is:

state	Cust_num
Roraima	45
Amapá	67
Acre	77
Amazonas	143
Rondônia	240

The states with low numbers of registered customers on Olist suggest that Olist should focus on penetrating these states in its future growth strategies.

```
%%sql
WITH
  states_prefixes AS (
    SELECT
      customer_state AS City,
      COUNT(DISTINCT customer_unique_id) Cust_num
    FROM
      customers
    GROUP BY
      1
    ORDER BY
      2 desc
    LIMIT
      5
  )
SELECT
  s.state,
  Cust_num
FROM
  states s
  JOIN states_prefixes sp ON s.prefix = sp.city
```

WHAT IS THE PERCENTAGE OF CUSTOMERS THAT DIDN'T MAKE ANY ORDERS SINCE REGISTERATION?

The result of this query Is:

Count	Percentage
676	0.70

As observed, only 0.7% of our customers did not make any purchases, indicating a very low rate which is positive for Olist.

```
%%sql
WITH
  inactive_cust AS (
    SELECT
      c.customer_Unique_id AS unique_id,
      SUM(price + freight_value) AS total_sales
    FROM
      customers c
      LEFT JOIN orders o ON c.customer_id = o.customer_id
      LEFT JOIN order_items oi ON o.order_id = oi.order_id
    GROUP BY
      1
    HAVING
      total_sales IS NULL
  )
SELECT
  COUNT(unique_id) AS `Count`,
  ROUND(
    COUNT(unique_id) * 1.00 / (
      SELECT
        COUNT(DISTINCT customer_unique_id)
      FROM
        customers
    ) * 100,
    2
  ) AS Percentage
FROM
  inactive_cust;
```

WHO ARE OUR TOP 100 CUSTOMERS BASED ON TOTAL SALES?

The result is the IDs of the top 100 customers, and here are the top five methods to engage and retain them effectively:

- Personalized communication through tailored emails and messages.
- Implementing an exclusive loyalty program offering rewards and benefits.
- Providing VIP treatment and exclusive access to products and events.
- Offering dedicated support via specialized customer service or personal account managers.
- Sending personalized gifts and surprises to reinforce their loyalty.

```
%%sql
SELECT
  c.customer_Unique_id,
  SUM(price + freight_value) AS total_sales
FROM
  customers c
  LEFT JOIN orders o ON c.customer_id = o.customer_id
  LEFT JOIN order_items oi ON o.order_id = oi.order_id
GROUP BY
  1
ORDER BY
  total_sales desc
LIMIT
  100;
```

02

Products Analysis

- How many product categories do we have?
- How many products does each category have?
- What are the top 20 products contributing to sales?
- What are the top 20 product categories contributing to sales?
- What are the top 3 product categories contributing to states' sales?

HOW MANY PRODUCTS DOES EACH CATEGORY HAVE?

This query will generate the most varied product category displayed on Olist:

product_category_name_english	products_per_category
bed_bath_table	3029
sports_leisure	2867
furniture_decor	2657
health_beauty	2444
housewares	2335
auto	1900
computers_accessories	1639
toys	1411
watches_gifts	1329

```
%%sql
WITH
  portugese_names AS (
    SELECT
      product_category_name,
      COUNT(product_category_name) AS products_per_category
    FROM
      products
    GROUP BY
      1
  )
SELECT
  ct.product_category_name_english,
  o.products_per_category
FROM
  portugese_names AS o
  INNER JOIN category_translation AS ct
  ON o.product_category_name = ct.product_category_name
ORDER BY
  2 DESC
```

WHAT ARE THE TOP 20 PRODUCT CATEGORIES CONTRIBUTING TO SALES?

The outcome of this query is the top 20 categories based on their sales contribution.

product_category_name_english	Total_Sales
health_beauty	1384784
watches_gifts	1285041
bed_bath_table	1151658
sports_leisure	1084445
computers_accessories	936675
furniture_decor	778022
housewares	702968
cool_stuff	702769
auto	651009

```
%%sql
SELECT
  product_category_name_english,
  ROUND(SUM(price + freight_value)) AS Total_Sales
FROM
  order_items oi
  JOIN products p
  ON oi.product_id = p.product_id
  JOIN category_translation ct
  ON p.product_category_name = ct.product_category_name
GROUP BY
  1
ORDER BY
  2 desc
LIMIT
  20;
```

WHAT ARE THE TOP 3 PRODUCT CATEGORIES CONTRIBUTING TO STATES' SALES?

This query will identify the top 3 product category sales in each state, illustrating how location influences category performance and reflects state preferences.

state	product_category_name_english	Total_Sales	Ranking
Acre	sports_leisure	2072	1
Acre	furniture_decor	1747	2
Acre	health_beauty	1655	3
Alagoas	health_beauty	14870	1
Alagoas	watches_gifts	12763	2
Alagoas	computers_accessories	9022	3
Amapá	computers_accessories	2443	1
Amapá	watches_gifts	1847	2
Amapá	health_beauty	1719	3

```
%%sql
WITH
  ranks AS (
    SELECT
      s.state,
      product_category_name_english,
      ROUND(SUM(price + freight_value)) AS Total_Sales,
      ROW_NUMBER() OVER (
        PARTITION BY
          s.state
        ORDER BY
          s.state,
          ROUND(SUM(price + freight_value)) desc
      ) AS Ranking
    FROM
      order_items oi
      JOIN products p ON oi.product_id = p.product_id
      JOIN category_translation ct
      ON p.product_category_name = ct.product_category_name
      JOIN orders o ON o.order_id = oi.order_id
      JOIN customers c ON c.customer_id = o.customer_id
      JOIN states s ON s.prefix = c.customer_state
    GROUP BY
      1,
      2
  )
SELECT
  *
FROM
  ranks
WHERE
  Ranking < 4
ORDER BY
  state,
  Ranking;
```

03

Sellers Analysis

- How many sellers are registered with OList?
- How are our sellers geographically distributed?
- Who are our top 50 sellers based on total sales?
- Who are our worst performing sellers based on total sales?

WHO ARE OUR TOP 50 SELLERS BASED ON TOTAL SALES?

Here are five key strategies for effectively engaging and retaining the top 50 sellers:

1. Training & Resources: Offer sellers comprehensive training materials for optimal performance.
2. Competitive Incentives: Provide attractive commission rates or discounts to motivate sellers.
3. Networking Opportunities: Facilitate seller connections and collaborations for community support.
4. Efficient Payments: Ensure timely and transparent payment processes to build trust.
5. Clear Communication: Establish easy-to-access channels for feedback and support.

```
%%sql
SELECT
  s.seller_id,
  SUM(price + freight_value) AS total_sales
FROM
  sellers s
  INNER JOIN order_items oi ON s.seller_id = oi.seller_id
GROUP BY
  1
ORDER BY
  2 desc
LIMIT
  50;
```

WHO ARE OUR WORST PERFORMING SELLERS BASED ON TOTAL SALES?

Here are five crucial approaches to effectively engage and retain the bottom-performing 50 sellers:

- Feedback and Coaching: Provide constructive feedback and coaching to address weaknesses and establish improvement plans.
- Goal Setting: Collaborate on setting clear, attainable goals to guide and inspire improvement.
- Regular Monitoring and Support: Maintain consistent communication, monitor progress, and offer assistance throughout the enhancement journey.
- Incentives: Offer rewards or incentives for achieving performance targets to drive seller motivation.
- Reassignment or Termination: Explore reassignment to different roles or, as a last resort, consider termination if performance does not improve despite support.

```
%%sql
SELECT
  s.seller_id,
  AVG(review_score)
FROM
  sellers s
  INNER JOIN order_items oi ON s.seller_id = oi.seller_id
  INNER JOIN orders o ON oi.order_id = o.order_id
  INNER JOIN order_reviews ov ON o.order_id = ov.order_id
GROUP BY
  1
ORDER BY
  2
LIMIT
  50
```

04

Orders Analysis

- What is the period of time covered in the data?
- How many orders were delayed, and what is their percentage?
- What is the average order value (AOV)?
- What is the average freight cost?
- What is the distribution of payment types?
- What is the distribution of sales along months?

HOW MANY DELAYED ORDERS, WHAT IS THEIR PERCENTAGE?

The result of the query is as follows:

Count	delayed_orders_percentage
7827	7.87

In the results, it is noted that the percentage of delayed orders is around 8%, which is relatively low. However, it is worth investigating further to determine if there is room for improvement.

```
%%sql
SELECT
  COUNT(*) AS `Count`,
  ROUND(
    COUNT(*) * 1.00 / (
      SELECT
        COUNT(*)
      FROM
        orders
    ) * 100,
    2
  ) AS delayed_orders_percentage
FROM
  orders
WHERE
  order_delivered_customer_date > order_estimated_delivery_date
```

WHAT IS THE DISTRIBUTION OF PAYMENT TYPES?

The result of the query is as follows:

payment_type	count	total_sales
credit_card	76795	12542084
boleto	19784	2869361
voucher	5775	379437
debit_card	1529	217990

The credit card is the preferred payment method in Olist. And Boleto comes second which is an official (regulated by the Central Bank of Brazil) payment method in Brazil. To complete a transaction, customers receive a voucher stating the amount to pay for services or goods.

```
%%sql
SELECT
  payment_type,
  COUNT(payment_type) AS `count`,
  ROUND(SUM(payment_value)) AS total_sales
FROM
  order_payments
GROUP BY
  1
HAVING
  total_sales
```

HOW MUCH IS THE AVERAGE ORDER VALUE (AOV)?

The result of the query is as follows:

avg_order_value
161

```
%%sql
WITH
  total_value AS (
    SELECT
      o.order_id,
      SUM(payment_value) AS order_value
    FROM
      orders o
      INNER JOIN order_payments op
      ON o.order_id = op.order_id
    GROUP BY
      1
  )
SELECT
  ROUND(AVG(order_value)) AS avg_order_value
FROM
  total_value
```

WHAT IS THE DISTRIBUTION OF SALES ALONG MONTHS OF THE STUDY PERIOD??

The result of the query is as follows:

YEAR	MONTH	Total_Sales
2016	9	219
2016	10	56945
2016	12	20
2017	1	92198
2017	2	282732
2017	3	396542
2017	4	352846
2017	5	586222
2017	6	543623

Due to incomplete data, it is challenging to determine whether there is seasonality or consistent rates based on the query's result set.

```sql
%%sql
WITH
  month_extraction AS (
    SELECT
      EXTRACT(
        YEAR
        FROM
          shipping_limit_date
      ) AS YEAR,
      EXTRACT(
        MONTH
        FROM
          shipping_limit_date
      ) AS MONTH,
      price,
      freight_value
    FROM
      order_items
  )
SELECT
  YEAR,
  MONTH,
  ROUND(SUM(price + freight_value)) AS Total_Sales
FROM
  month_extraction
GROUP BY
  1,
  2
ORDER BY
  1,
  2;
```

CONCLUSION

In conclusion, our SQL analysis has provided comprehensive insights across key business areas. We've identified customer distribution by state and city, pinpointed top and bottom performers, and analyzed product categories and seller geography. Additionally, we've examined critical order metrics such as delays, average order value, and payment type distribution. These insights offer actionable intelligence to optimize strategies, enhance customer engagement, and streamline operations for sustained growth.