Table Of Contents

Q1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

i. Data type of columns in a table.

Out of 8 tables I have selected 4 tables that will give me a clear picture to perform the analysis to reach a meaningful outcome.

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
product_id	STRING	NULLABLE				
product_category	STRING	NULLABLE				
product_name_length	INTEGER	NULLABLE				
product_description_length	INTEGER	NULLABLE				
product_photos_qty	INTEGER	NULLABLE				
product_weight_g	INTEGER	NULLABLE				
product_length_cm	INTEGER	NULLABLE				
product_height_cm	INTEGER	NULLABLE				
product_width_cm	INTEGER	NULLABLE				

```
SELECT * from `business_case.products` limit 10;
```

products

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
customer_id	STRING	NULLABLE				
customer_unique_id	STRING	NULLABLE				
customer_zip_code_prefix	INTEGER	NULLABLE				
customer_city	STRING	NULLABLE				
customer_state	STRING	NULLABLE				

```
SELECT * from `business_case.customers` limit 10;
```

customers

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
order_id	STRING	NULLABLE				
customer_id	STRING	NULLABLE				
order_status	STRING	NULLABLE				
order_purchase_timestamp	TIMESTAMP	NULLABLE				
order_approved_at	TIMESTAMP	NULLABLE				
order_delivered_carrier_date	TIMESTAMP	NULLABLE				
order_delivered_customer_date	TIMESTAMP	NULLABLE				
order_estimated_delivery_date	TIMESTAMP	NULLABLE				

```sql
SELECT * FROM `business_case.orders` LIMIT 10;
```

orders

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
product_id	STRING	NULLABLE				
product_category	STRING	NULLABLE				
product_name_length	INTEGER	NULLABLE				
product_description_length	INTEGER	NULLABLE				
product_photos_qty	INTEGER	NULLABLE				
product_weight_g	INTEGER	NULLABLE				
product_length_cm	INTEGER	NULLABLE				
product_height_cm	INTEGER	NULLABLE				
product_width_cm	INTEGER	NULLABLE				

```sql
SELECT * FROM `business_case.products` LIMIT 10;
```

products

---

ii. Time period for which the data is given

2016-2018 is our time period.

Below query to find out the timeline between the orders

```
SELECT
MIN(order_purchase_timestamp) AS first_order,
MAX(order_purchase_timestamp) AS last_order,
FROM `business_case.orders`;
```

Time_Line

iii. Cities and States of customers ordered during the given period

```
SELECT

customer_unique_id,

customer_city,

customer_state

FROM `business_case.customers`

GROUP BY customer_unique_id, customer_city,customer_state

LIMIT 10;
```

City&State_Orders

Q2. In-depth Exploration:

i.Is there a growing trend on e-commerce in Brazil? How can we describe complete scenario? Can we see some seasonality with peaks at specific Months?

The 3 questions can be answered by using the query mentioned below.
Seasonality = max orders placed in a specific month

```
WITH months as(
  SELECT
    EXTRACT(month from order_purchase_timestamp) as month,
    COUNT (*) as no_of_orders
  From `business_case.orders`
  GROUP BY month
)
SELECT
  month,
  no_of_orders,
  RANK() OVER(ORDER BY no_of_orders DESC) as rank
FROM months
order by rank;
```

ii. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon Night)?

With the help of the below query

```
WITH time_of_the_day AS(
  SELECT
  EXTRACT(HOUR FROM order_purchase_timestamp) as hour,
  COUNT(*) AS count_of_logins
  FROM `business_case.orders`
  GROUP BY hour
 )

 SELECT
  hour,
  count_of_logins,
  CASE
    WHEN hour BETWEEN 0 and 5 THEN 'Dawn'
    WHEN hour BETWEEN 6 and 11 THEN 'Morning'
```

```
      WHEN hour BETWEEN 12 and 17 THEN 'Afternoon'
      WHEN hour BETWEEN 18 and 23 THEN 'Night'
    END AS time_of_day
    FROM time_of_the_day
    ORDER BY time_of_the_day.HOUR
```

Time_of_day_1

The grouping logins i.e the sum of total logins as per the user

```
 WITH time_of_day as(
    SELECT
        EXTRACT(HOUR FROM order_purchase_timestamp) as hour,
    FROM `business_case.orders`
)

SELECT
    CASE
    WHEN hour BETWEEN 0 and 5 THEN 'Dawn'
        WHEN hour BETWEEN 6 and 11 THEN 'Morning'
        WHEN hour BETWEEN 12 and 17 THEN 'Afternoon'
        WHEN hour BETWEEN 18 and 23 THEN 'Night'
    END AS time_of_day,
    COUNT(*) as count_of_logins
FROM time_of_day
GROUP BY time_of_day
ORDER BY count_of_logins DESC;
```

Time_of_Day_2

Q3. Evolution of E-commerce orders in the Brazil region:

i. Get month on month orders by states

Below query to get month on month orders.

```
WITH month_sales AS(
  SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) as month,
    SUM(order_item.price) as sales
  FROM
    `business_case.orders` as orders
  JOIN
    `business_case.order_items` as order_item
  ON
    orders.order_id = order_item.order_id
  GROUP BY
    month
)
SELECT
  month,
  sales,
  LAG(sales) OVER(ORDER BY month) as prev_month_sales,
  sales - LAG(sales) OVER(ORDER BY month) as month_on_month_sales
FROM
  month_sales
ORDER BY
  month ASC
```

Month_Sales

However, we need the same data state wise. So we will be considering the parameter of delivered orders as a marker of confirmed sales.

The table considered for this exploration are tables orders, customers and geolocation to get the answer.

```
WITH orders_customers AS (
SELECT
 o.order_id,
 o.customer_id,
```

```
 o.order_purchase_timestamp,
 o.order_status,
 c.customer_zip_code_prefix
FROM
 `business_case.orders` o
JOIN
 `business_case.customers` c
ON
 o.customer_id = c.customer_id
),
geolocated_orders AS (
SELECT
 oc.order_id,
 oc.customer_id,
 oc.order_purchase_timestamp,
 oc.order_status,
 g.geolocation_zip_code_prefix,
 g.geolocation_city,
 g.geolocation_state
FROM
 orders_customers oc
JOIN
 `business_case.geolocation` g
ON
 oc.customer_zip_code_prefix = g.geolocation_zip_code_prefix
),
month_on_month AS (
SELECT
 EXTRACT(MONTH FROM order_purchase_timestamp) as month,
 EXTRACT(YEAR FROM order_purchase_timestamp) as year,
 geolocation_state,
 COUNT(DISTINCT order_id) as sales
FROM
 geolocated_orders
WHERE
 order_status = 'delivered'
GROUP BY
 month,
 year,
 geolocation_state
ORDER BY
```

```
 year,
 month,
 geolocation_state
)
SELECT
month,
year,
geolocation_state,
sales,
LAG(sales) OVER (PARTITION BY geolocation_state ORDER BY year, month) as
prev_month_sales
FROM
month_on_month;
```

State_wise_month_sales

ii. Distribution of customers across the states in Brazil
We have to check customer distribution across states in Brazil by fetching the data via BigQuery:

```
SELECT
  customer_state,
  COUNT(DISTINCT customer_id) as customers
FROM
  `business_case.customers`
GROUP BY
  customer_state
ORDER BY
  customers DESC;
```

Distribution_of_customers

Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table.

```
 WITH orders_payments AS (
  SELECT
    o.order_id,
    o.order_purchase_timestamp,
    p.payment_value
  FROM `business_case.orders` o
  JOIN `business_case.payments` p
  ON o.order_id = p.order_id
),

orders_payments_2017 AS(
  SELECT
    SUM(payment_value) as total_cost_2017
  FROM orders_payments
  WHERE
    EXTRACT(YEAR FROM order_purchase_timestamp) = 2017
    AND EXTRACT(MONTH FROM order_purchase_timestamp) >= 1
    AND EXTRACT(MONTH FROM order_purchase_timestamp) <= 8
),

orders_payments_2018 AS(
  SELECT
    SUM(payment_value) as total_cost_2018
  FROM orders_payments
  WHERE
    EXTRACT(YEAR FROM order_purchase_timestamp) = 2018
    AND EXTRACT(MONTH FROM order_purchase_timestamp) >= 1
    AND EXTRACT(MONTH FROM order_purchase_timestamp) <= 8
)

SELECT
  (total_cost_2018 - total_cost_2017)/total_cost_2017 * 100 AS
percentage_increase
FROM orders_payments_2017, orders_payments_2018;
```

Percentage_Increase

ii. Mean & Sum of price and freight value by customer state

```
WITH value AS (
SELECT
 c.customer_state AS customer_state,
 SUM(oi.price) as price_sum,
 AVG(oi.price) as price_avg,
 SUM(oi.freight_value) as freight_sum,
 AVG(oi.freight_value) as freight_avg
FROM
 `business_case.order_items` oi
JOIN
 `business_case.orders`o
 ON oi.order_id = o.order_id
JOIN
 `business_case.customers` c
 ON o.customer_id = c.customer_id
GROUP BY
 customer_state
)
SELECT
customer_state,
price_sum,
price_avg,
freight_sum,
freight_avg
FROM
value;
```

Mean&Sum

To get an idea of Target's real selling price I have further analyzed it by averaging Net Selling Price by different states(assumption: Target bears the shipping cost)

```
WITH value AS (
  SELECT
    c.customer_state AS customer_state,
```

```
    SUM(oi.price) AS price_sum,
    AVG(oi.price) AS price_avg,
    SUM(oi.freight_value) AS fright_sum,
    AVG(oi.freight_value) AS frieght_avg
  FROM
    `business_case.order_items` oi
  JOIN
    `business_case.orders` o
    ON oi.order_id = o.order_id
  JOIN
    `business_case.customers` c
    ON o.customer_id = c.customer_id
  GROUP BY
    customer_state
)

SELECT
  customer_state,
  price_sum,
  price_avg,
  frieght_sum,
  frieght_avg,
  price_avg - frieght_avg AS Net_SP
FROM
  value;
```

Avg_Net_SP

Q5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery. Find time_to_delivery & diff_estimated_delivery.

Formula for the same given below:

- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date - Order_delivered_customer_date

```
WITH delivery_data AS (
SELECT
  order_id,
  customer_id,
  order_status,
  order_purchase_timestamp,
  order_delivered_carrier_date,
  order_delivered_customer_date,
  order_estimated_delivery_date,
DATE_DIFF(order_delivered_carrier_date, order_purchase_timestamp, DAY) AS
days_between_purchase_delivery,
DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) AS
days_between_estimated_actual_delivery
FROM
  `business_case.orders`
)
SELECT
  order_id,
  customer_id,
  order_status,
  order_purchase_timestamp,
  order_delivered_carrier_date,
  order_delivered_customer_date,
  order_estimated_delivery_date,
  days_between_purchase_delivery,
  days_between_estimated_actual_delivery,
FROM
  Delivery_data;
```

Analysis sales,freight&delivery time

ii. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery.

```sql
WITH sample AS (
SELECT
o.order_id,
oi.freight_value,
c.customer_state
FROM
`business_case.orders` o
JOIN
`business_case.order_items` oi
ON
o.order_id = oi.order_id
JOIN
`business_case.customers` c
ON
o.customer_id = c.customer_id
),
delivery_data AS (
SELECT
o.order_id,
o.customer_id,
DATE_DIFF(o.order_delivered_carrier_date, o.order_purchase_timestamp, DAY) AS
days_between_purchase_delivery,
DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date,
DAY) AS days_between_estimated_actual_delivery
FROM
`business_case.orders` o
)
SELECT
sample.customer_state,
AVG(sample.freight_value) as avg_freight_value,
AVG(delivery_data.days_between_purchase_delivery) as
avg_days_between_purchase_delivery,
AVG(delivery_data.days_between_estimated_actual_delivery) as
avg_days_between_estimated_actual_delivery
FROM
sample
JOIN delivery_data
ON sample.order_id=delivery_data.order_id
GROUP BY
sample.customer_state
```

Est_Delivery

iii. Sort the data to get the following:

iii(i). Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

In DESC Order:

```
WITH value AS(
  SELECT
    o.order_id,
    oi.freight_value,
    c.customer_state
  FROM
    `business_case.orders` o
  JOIN
    `business_case.order_items` oi
  ON
    o.order_id = oi.order_id
  JOIN
    `business_case.customers` c
  ON
    o.customer_id = c.customer_id
)

SELECT
  customer_state,
  AVG(freight_value) as avg_freight_value
FROM
  value
GROUP BY
  customer_state
ORDER BY
  avg_freight_value DESC
LIMIT 5;
```

avg_freight_val_desc

In ASC Order:

```
WITH value AS(
  SELECT
```

```
    o.order_id,
    oi.freight_value,
    c.customer_state
  FROM
    `business_case.orders` o
  JOIN
    `business_case.order_items` oi
  ON
    o.order_id = oi.order_id
  JOIN
    `business_case.customers` c
  ON
    o.customer_id = c.customer_id
)

SELECT
  customer_state,
  AVG(freight_value) as avg_freight_value
FROM
  value
GROUP BY
  customer_state
ORDER BY
  avg_freight_value ASC
LIMIT 5;
```

avg_freight_val_asc

iii(ii).Top 5 states with highest/lowest average time to delivery

For highest average time to delivery:

```
WITH delivery_data AS (
SELECT
o.order_id,
o.customer_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_delivery,
c.customer_state
FROM
`business_case.orders` o
```

```
JOIN
`business_case.customers` c
ON o.customer_id = c.customer_id
)
SELECT
customer_state,
AVG(time_to_delivery) as avg_time_to_delivery
FROM
delivery_data
GROUP BY
customer_state
ORDER BY
avg_time_to_delivery DESC
LIMIT 5;
```

avg_del_time_desc

For the lowest average time to delivery:

```
WITH delivery_data AS (
SELECT
o.order_id,
o.customer_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_delivery,
c.customer_state
FROM
`business_case.orders` o
JOIN
`business_case.customers` c
ON o.customer_id = c.customer_id
)
SELECT
customer_state,
AVG(time_to_delivery) as avg_time_to_delivery
FROM
delivery_data
GROUP BY
customer_state
ORDER BY
avg_time_to_delivery ASC
LIMIT 5;
```

avg_del_time_asc

iii(iii). Top 5 states where delivery is really fast/ not so fast compared to estimated date

For the Fastest:

```
WITH delivery_data AS (
SELECT
o.order_id,
oi.freight_value,
c.customer_state,
DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date,
DAY) AS days_between_estimated_actual_delivery
FROM
`business_case.orders` o
JOIN
`business_case.order_items` oi
ON
o.order_id = oi.order_id
JOIN
`business_case.customers` c
ON
o.customer_id = c.customer_id
)
SELECT
customer_state,
AVG(days_between_estimated_actual_delivery) as
avg_days_between_estimated_actual_delivery
FROM
delivery_data
GROUP BY
customer_state
ORDER BY
avg_days_between_estimated_actual_delivery ASC
LIMIT 5
```

fastest_delivery

For the Slowest:

```
WITH delivery_data AS (
```

```
SELECT
o.order_id,
oi.freight_value,
c.customer_state,
DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date,
DAY) AS days_between_estimated_actual_delivery
FROM
`business_case.orders` o
JOIN
`business_case.order_items` oi
ON
o.order_id = oi.order_id
JOIN
`business_case.customers` c
ON
o.customer_id = c.customer_id
)
SELECT
customer_state,
AVG(days_between_estimated_actual_delivery) as
avg_days_between_estimated_actual_delivery
FROM
delivery_data
GROUP BY
customer_state
ORDER BY
avg_days_between_estimated_actual_delivery DESC
LIMIT 5;
```

slowest_delivery

Q6. Payment type analysis:

i.Month over Month count of orders for different payment types

```
WITH orders_data AS (
SELECT
o.order_id,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month, payment_type
FROM
`business_case.orders` o
JOIN
`business_case.payments` p
ON
o.order_id = p.order_id
)
SELECT
month,
payment_type,
COUNT(DISTINCT order_id) AS order_count
FROM
orders_data
GROUP BY
month,
payment_type
ORDER BY
month,
payment_type;
```

M_O_M_Diff_Payment_Types

ii.Count of orders based on the no. of payment installments

```
WITH orders_payments AS (
SELECT
o.order_id,
o.order_purchase_timestamp,
p.payment_installments,
p.payment_value
FROM
`business_case.orders` o
JOIN
`business_case.payments` p
ON
```

```
o.order_id = p.order_id
)
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) as month, payment_installments,
COUNT(order_id) as order_count,
FROM
orders_payments
GROUP BY
month,
payment_installments
ORDER BY
month,
Payment_installments;
```

no_of_installments

Q7. Actionable Insights

- Brazil is one of the 8th largest economies in the world and it is also reflected in Sales in Brazil the peak sales in Brazil are in months of June, July and August. Especially the festival of Festa Junina which is big festival as majority of the population follow Catholic tradition which goes on from June till end of July followed by Festival de Cachaça which is held over a long weekend of August as shown in Month on Month Sales.

- Brazilian customers prefer the afternoon time to shop as most of the working class have their lunch breaks, kids come home from school, housewives finish their day chores. The main reason for this is to spend their free time to shop for the things that they need for the future and also to update themselves on what is going on in the world.

- From the data, it appears that the state of Sao Paulo (SP) has the highest number of online customers, with 41,746 customers. The state of Rio de Janeiro (RJ) is in second place with 12,852 customers. The state of Minas Gerais (MG) is in third place with 11,635 customers. The states of Santa Catarina (SC), Parana (PR), and Bahia (BA) do have relatively lesser numbers of online customers. The state of Sao Paulo has much better stats compared to its peers. The state of Roraima has less online shoppers compared to other states in Brazil as its population is lesser compared to other states, and also it could be due to low accessibility of modern technology to access the websites, as this is just my inference to be 99% sure more data is required.

- We see that there was a 136.97% rise in the cost of orders from 2017 to 2018 (from January to August). It determines that there was a significant increase during this period especially the cost of goods. During this period there were elections because the new government came in charge with new policies such as promoting privatization, reform in the pension system and deregulation of economic activity.

- Being the capital of Brazil, Sao Paulo(SP) the state with the highest average price per order with an average of 109.6536292.The state with the largest sum of freight cost is Sao Paulo, with a total of 718723.07.The state with the highest average freight cost per order is Tocantins(TO), with an average of 37.24660317. Roraima is the state with the lowest number of customers, with a total of 7829.43 in price sum. The state with

the lowest average price per order is Acre, with an average of 173.7277174. The state with the lowest sum of freight cost is Roraima, with a total of 2235.19.The state with the lowest average freight cost per order is Amapá(AM), with an average of 34.00609756. RR and AC have the lowest average order value, indicating that customers in these states tend to place lower-value orders, it also may be the case that the state has a lesser population.

- The Delivery time analysis, gives us the most of the cases delivered before ETA. This might also indicate that the sellers are either not providing accurate ETA or underestimating the logistics reliability of the country. This can also be an indicator of sellers being fearful of the contingencies.

- Moreover, it can also be inferred that the shipping performance varies significantly among different states and the company may want to investigate the reasons for such kinds of variations and take appropriate actions to improve its performance in these states.

- To have a quick glance of this, the states of PB,RR and RO have the highest average freight values, while the states of SP, PR, and MG have the lowest average freight value. In terms of time to delivery, the states of RR, AP, and AM have the highest average time to delivery, while the states of SP, PR, and MG have the lowest average time to delivery.

- In terms of delivery speed compared to the estimated date, the states of AC, RO, and AM have the fastest delivery, while the states of Alagoas(AL), Maranhao, and Sergipe(SE) have the slowest delivery.

- The Payment Analysis i.e Month on Month Basis shows steady increase in use of UPI and Credit Card. It shows loopholes in payment data and needs to be researched ahead. The voucher is not as widely accepted as the others.

Q8. Recommendations

- Based on the actionable insights provided, kindly view the recommendations that can be made for the company:

- Promotions such as promo codes, refer and post it on social media platforms will improve the revenue by a good margin, as Brazilian customers tend to be active during afternoon food coupons and festival passes may also do the trick

- Focusing more on increasing sales in states with lower sales, such as AM, also to further investigate the reasons for the lower sales in these states.

- The inventory should be stocked more in the months of May, July, and August to plan stock and promotions accordingly.

- Utilizing the knowledge of peak selling months (May, July, and August) to plan inventory and promotions accordingly.

- States like RR where both delivery time is high, order count and value is low could be an indicator for the company to not divert resources to that state and consider a 2 year pause and revisit the option to open up the customers again.

- For Other States with higher delivery time the company needs to either incorporate more local sellers from those states or scale up its warehousing capacities in those states.

- Sao Paulo is a key market for the company, considering expanding efforts to attract and retain customers in this state.

- Investigate the reasons for the significant increase in the cost of orders from 2017 to 2018 and take appropriate action to mitigate these factors.

- Investigate the variations in shipping performance among different states and take appropriate actions to improve performance in these states.

- Monitor the trend of increased use of UPI and credit card payments and consider implementing additional payment options for customers.

- Reduction in the use of vouchers as a payment approach is a good step.