

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

a. Data type of columns in a table

orders QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

| <input type="checkbox"/> | Field name | Type | Mode | Key | Collation | Default Value | Policy Tags | Description |
|--------------------------|---|-----------|----------|-----|-----------|---------------|-------------|-------------|
| <input type="checkbox"/> | order_id | STRING | NULLABLE | | | | | |
| <input type="checkbox"/> | customer_id | STRING | NULLABLE | | | | | |
| <input type="checkbox"/> | order_status | STRING | NULLABLE | | | | | |
| <input type="checkbox"/> | order_purchase_timestamp | TIMESTAMP | NULLABLE | | | | | |
| <input type="checkbox"/> | order_approved_at | TIMESTAMP | NULLABLE | | | | | |
| <input type="checkbox"/> | order_delivered_carrier_date | TIMESTAMP | NULLABLE | | | | | |
| <input type="checkbox"/> | order_delivered_customer_date | TIMESTAMP | NULLABLE | | | | | |
| <input type="checkbox"/> | order_estimated_delivery_date | TIMESTAMP | NULLABLE | | | | | |

Different data types of columns for orders table

- b. Time period for which the data is given

```
select min(order_purchase_timestamp),
       max(order_purchase_timestamp)
from `target.orders`
```

Untitled 7 RUN SAVE SHARE SCHEDULE MORE

```
1 select min(order_purchase_timestamp),
2       max(order_purchase_timestamp)
3 from `target.orders`
4
```

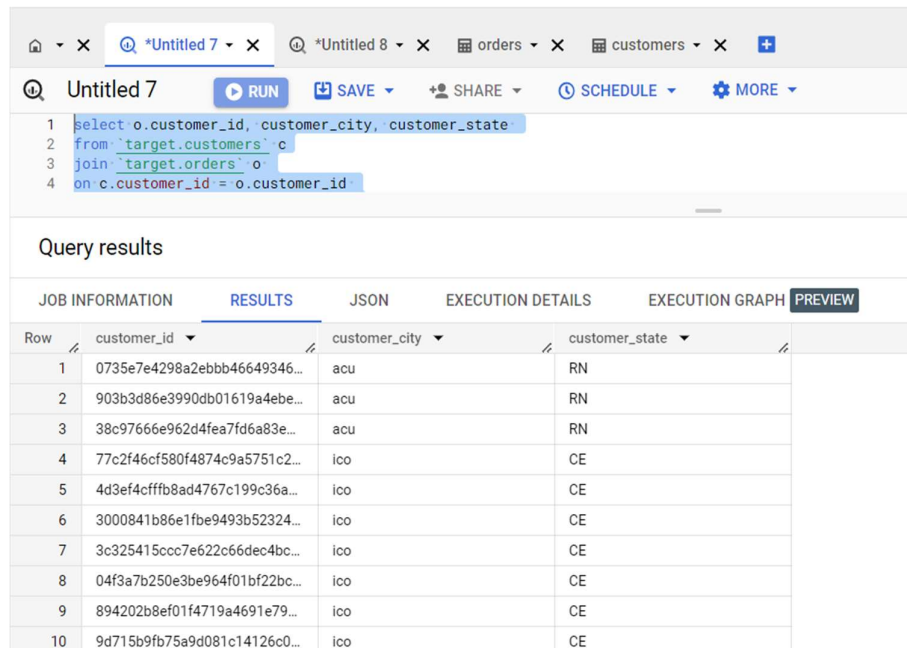
Query results

| Row | f0_ | f1_ |
|-----|-------------------------|-------------------------|
| 1 | 2016-09-04 21:15:19 UTC | 2018-10-17 17:30:18 UTC |

Time period for the given data is between 4 Apr 2016 and 17 Oct 2018

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

```
select o.customer_id, customer_city, customer_state
from `target.customers` c
join `target.orders` o
on c.customer_id = o.customer_id
where o.order_purchase_timestamp between '2016-09-04' and '2018-10-17'
```



The screenshot shows a SQL query editor with the following query:

```
1 select o.customer_id, customer_city, customer_state
2 from `target.customers` c
3 join `target.orders` o
4 on c.customer_id = o.customer_id
```

Below the query, the "Query results" section is displayed with tabs for "JOB INFORMATION", "RESULTS", "JSON", "EXECUTION DETAILS", and "EXECUTION GRAPH". The "RESULTS" tab is active, showing a table with 10 rows of data.

| Row | customer_id | customer_city | customer_state |
|-----|-------------------------------|---------------|----------------|
| 1 | 0735e7e4298a2ebbb46649346... | acu | RN |
| 2 | 903b3d86e3990db01619a4ebe... | acu | RN |
| 3 | 38c97666e962d4fea7fd6a83e... | acu | RN |
| 4 | 77c2f46cf580f4874c9a5751c2... | ico | CE |
| 5 | 4d3ef4cfff8ad4767c199c36a... | ico | CE |
| 6 | 3000841b86e1f8e9493b52324... | ico | CE |
| 7 | 3c325415ccc7e622c66dec4bc... | ico | CE |
| 8 | 04f3a7b250e3be964f01bf22bc... | ico | CE |
| 9 | 894202b8ef01f4719a4691e79... | ico | CE |
| 10 | 9d715b9fb75a9d081c14126c0... | ico | CE |

2. In-depth Exploration:

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

```
select
extract(year from order_purchase_timestamp) as year,
count(order_id) as no_of_orders
from `target.orders`
group by 1
order by 1
```

| Row | year | no_of_orders |
|-----|------|--------------|
| 1 | 2016 | 329 |
| 2 | 2017 | 45101 |
| 3 | 2018 | 54011 |

From the above picture we can say that there is growing trend on e-commerce in brazil.

Since there is inadequate information on all the months of year 2016 it is difficult to predict year over year change in trend.

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(order_id) as no_of_orders
from `target.orders`
group by 1, 2
order by 1,2
```

| Row | year | month | no_of_orders | Row | year | month | no_of_orders | Row | year | month | no_of_orders |
|-----|------|-------|--------------|-----|------|-------|--------------|-----|------|-------|--------------|
| 1 | 2016 | 9 | 4 | 10 | 2017 | 7 | 4026 | 16 | 2018 | 1 | 7269 |
| 2 | 2016 | 10 | 324 | 11 | 2017 | 8 | 4331 | 17 | 2018 | 2 | 6728 |
| 3 | 2016 | 12 | 1 | 12 | 2017 | 9 | 4285 | 18 | 2018 | 3 | 7211 |
| 4 | 2017 | 1 | 800 | 13 | 2017 | 10 | 4631 | 19 | 2018 | 4 | 6939 |
| 5 | 2017 | 2 | 1780 | 14 | 2017 | 11 | 7544 | 20 | 2018 | 5 | 6873 |
| 6 | 2017 | 3 | 2682 | 15 | 2017 | 12 | 5673 | 21 | 2018 | 6 | 6167 |
| 7 | 2017 | 4 | 2404 | 16 | 2018 | 1 | 7269 | 22 | 2018 | 7 | 6292 |
| 8 | 2017 | 5 | 3700 | 17 | 2018 | 2 | 6728 | 23 | 2018 | 8 | 6512 |
| 9 | 2017 | 6 | 3245 | 18 | 2018 | 3 | 7211 | 24 | 2018 | 9 | 16 |
| 10 | 2017 | 7 | 4026 | 19 | 2018 | 4 | 6939 | 25 | 2018 | 10 | 4 |

From the above pictures we can see the number of orders placed per month.

It can be seen that the peak orders were placed during end of year in 2017 and in 2018 there has been consistency in orders that have been placed.

In the year 2016 there have been months where no order has been placed but as the time went no of orders kept on increasing.

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

```
select sum(case when a.hour between 0 and 5 then a.no_of_orders end) as Dawn,
sum(case when a.hour between 6 and 11 then a.no_of_orders end) as
Morning,
sum(case when a.hour between 12 and 17 then a.no_of_orders end) as
Afternoon,
sum(case when a.hour between 18 and 24 then a.no_of_orders end) as Night
from
(select
extract(hour from order_purchase_timestamp) as hour,
count(order_id) as no_of_orders
from `target.orders`
group by 1
```

Query results

| JOB INFORMATION | | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW |
|-----------------|--------|-----------|-------------|-------------------|-----------------|---------|
| Row | Dawn ▼ | Morning ▼ | Afternoon ▼ | Night ▼ | | |
| 1 | 4740 | 22240 | 38361 | 34100 | | |

From the above picture, we can say that Brazilian customers buy mostly in afternoon or night.

Very few people tend to buy during dawn.

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

a. Get month on month orders by states

```
select a.customer_state,
       a.no_of_orders as monthly_orders,
       lead(a.no_of_orders) over(partition by a.customer_state order by a.month) as
next_month,
       round((lead(a.no_of_orders) over(partition by a.customer_state order by
a.month) - a.no_of_orders)/a.no_of_orders *100,2) as mom_orders
from
(select customer_state,
       extract(month from order_purchase_timestamp) as month,
       count(order_id) as no_of_orders
from `target.customers` c
left join `target.orders` o
on c.customer_id = o.customer_id
group by customer_state,month
order by customer_state,month)a
```

| JOB INFORMATION | | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW |
|-----------------|------------------|------------------|--------------|-------------------|-----------------|---------|
| Row | customer_state ▼ | monthly_orders ▼ | next_month ▼ | mom_orders ▼ | | |
| 1 | SE | 24 | 27 | 12.5 | | |
| 2 | SE | 27 | 43 | 59.26 | | |
| 3 | SE | 43 | 27 | -37.21 | | |
| 4 | SE | 27 | 19 | -29.63 | | |
| 5 | SE | 19 | 37 | 94.74 | | |
| 6 | SE | 37 | 42 | 13.51 | | |
| 7 | SE | 42 | 43 | 2.38 | | |
| 8 | SE | 43 | 16 | -62.79 | | |
| 9 | SE | 16 | 25 | 56.25 | | |

Here -ve sign in mom_orders represent decrease in percentage of number of orders compared to previous month.

b. Distribution of customers across the states in Brazil

```
select customer_state,
count (distinct customer_id) as no_of_customers
from `target.customers`
group by 1
```

| Row | customer_state | no_of_customers |
|-----|----------------|-----------------|
| 1 | RN | 485 |
| 2 | CE | 1336 |
| 3 | RS | 5466 |
| 4 | SC | 3637 |
| 5 | SP | 41746 |
| 6 | MG | 11635 |
| 7 | BA | 3380 |
| 8 | RJ | 12852 |
| 9 | GO | 2020 |

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

```
select a.month,
a.cost_of_orders as cost_2017,
lead(a.cost_of_orders,8) over(order by a.year,a.month) as cost_2018,
(lead(a.cost_of_orders,8) over(order by a.year,a.month)-
a.cost_of_orders)/a.cost_of_orders*100 as percentage_increase

from
(select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
round(sum(payment_value),0) as cost_of_orders
from `target.orders` o
join `target.payments` p
on o.order_id = p.order_id
where extract(month from order_purchase_timestamp) < 9
group by 1,2
order by 1,2)a
order by year,month
```

| Row | month | cost_2017 | cost_2018 | percentage_increase |
|-----|-------|-----------|-----------|---------------------|
| 1 | 1 | 138488.0 | 1115004.0 | 705.1267979897... |
| 2 | 2 | 291908.0 | 992463.0 | 239.9917097167... |
| 3 | 3 | 449864.0 | 1159652.0 | 157.7783507904... |
| 4 | 4 | 417788.0 | 1160785.0 | 177.8406751749... |
| 5 | 5 | 592919.0 | 1153982.0 | 94.62725937269... |
| 6 | 6 | 511276.0 | 1023880.0 | 100.2597422918... |
| 7 | 7 | 592383.0 | 1066541.0 | 80.04247252200... |
| 8 | 8 | 674396.0 | 1022425.0 | 51.60602969175... |
| 9 | 1 | 1115004.0 | null | null |

The percentage increase in cost from 2017 to 2018 for months Jan - Aug.
From the table we can see that there is a massive increase in cost of orders when compared to previous year.

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

```
select customer_state,
       round(avg(price)) as mean_of_price,
       round(sum(price)) as sum_of_price,
       round(avg(freight_value)) as mean_of_freight_value,
       round(sum(freight_value)) as sum_of_freight_value
from `target.order_items` oi
join `target.orders` o
on oi.order_id = o.order_id
join `target.customers` c
on o.customer_id = c.customer_id
group by 1
```

| customer_state | mean_of_price | sum_of_price | mean_of_freight_value | sum_of_freight_value |
|----------------|---------------|--------------|-----------------------|----------------------|
| SP | 110.0 | 5202955.0 | 15.0 | 718723.0 |
| RJ | 125.0 | 1824093.0 | 21.0 | 305589.0 |
| PR | 119.0 | 683084.0 | 21.0 | 117852.0 |
| SC | 125.0 | 520553.0 | 21.0 | 89660.0 |
| DF | 126.0 | 302604.0 | 21.0 | 50625.0 |
| MG | 121.0 | 1585308.0 | 21.0 | 270853.0 |
| PA | 166.0 | 178948.0 | 36.0 | 38699.0 |
| BA | 135.0 | 511350.0 | 26.0 | 100157.0 |
| GO | 126.0 | 294592.0 | 23.0 | 53115.0 |
| RS | 120.0 | 750304.0 | 22.0 | 135523.0 |

The mean price and mean freight value of state SP is lowest when compared to others.

The places where mean freight value is comparatively high we can maybe have more distributors across the state.

5. Analysis on sales, freight and delivery time

a. Calculate days between purchasing, delivering and estimated delivery

```
select a.order_id,
       date_diff(a.c,a.p,day) as days_bw_delivery_purchase,
       date_diff(a.e,a.p,day) as days_bw_estimateddel_purchase
from
```

```
(select order_id,
       extract(date from order_purchase_timestamp) as p,
       extract(date from order_delivered_customer_date) as c,
       extract(date from order_estimated_delivery_date) as e
from `target.orders`)a
where a.c is not null
```

| Row | order_id | days_bw_delivery_purchase | days_bw_estimateddel_purchase |
|-----|-------------------------------|---------------------------|-------------------------------|
| 1 | 1950d777989f6a877539f5379... | 30 | 18 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 31 | 60 |
| 3 | 65d1e226dfaeb8cdc42f66542... | 36 | 53 |
| 4 | 635c894d068ac37e6e03dc54e... | 31 | 33 |
| 5 | 3b97562c3aee8bdedcb5c2e45... | 33 | 34 |
| 6 | 68f47f50f04c4cb6774570cfde... | 30 | 32 |
| 7 | 276e9ec344d3bf029ff83a161c... | 44 | 40 |
| 8 | 54e1a3c2b97fb0809da548a59... | 41 | 37 |

b. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

```
select a.order_id,
       date_diff(a.c,a.p,day) as time_to_delivery,
       date_diff(a.e,a.p,day) as diff_estimated_delivery
from
```

```
(select order_id,
       extract(date from order_purchase_timestamp) as p,
       extract(date from order_delivered_customer_date) as c,
       extract(date from order_estimated_delivery_date) as e
from `target.orders`)a
where a.c is not null
```

| Row | order_id | time_to_delivery | diff_estimated_delivery |
|-----|-------------------------------|------------------|-------------------------|
| 1 | 1950d777989f6a877539f5379... | 30 | 18 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 31 | 60 |
| 3 | 65d1e226dfaeb8cdc42f66542... | 36 | 53 |
| 4 | 635c894d068ac37e6e03dc54e... | 31 | 33 |
| 5 | 3b97562c3aee8bdedcb5c2e45... | 33 | 34 |
| 6 | 68f47f50f04c4cb6774570cfde... | 30 | 32 |
| 7 | 276e9ec344d3bf029ff83a161c... | 44 | 40 |
| 8 | 54e1a3c2b97fb0809da548a59... | 41 | 37 |

Load more

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

```
select customer_state as state,
       round(avg(freight_value)) as mean_freight_value,
       round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)))
as time_to_delivery,
       round(avg(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day))) as
diff_estimated_delivery
from `target.order_items` oi
join `target.orders` o
on oi.order_id = o.order_id
join `target.customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
```

- d. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Top 5 states with highest avg freight value

```
select a.state,
       a.mean_freight_value as top_five
from
(select customer_state as state,
       round(avg(freight_value)) as mean_freight_value

from `target.order_items` oi
join `target.orders` o
on oi.order_id = o.order_id
join `target.customers` c
on o.customer_id = c.customer_id
```



```
group by c.customer_state)a
order by a.mean_freight_value desc
limit 5
```

Top 5 states with lowest avg freight value

```
select a.state,
       a.mean_freight_value as top_five_lowest
from
(select customer_state as state,
       round(avg(freight_value)) as mean_freight_value

from `target.order_items` oi
join `target.orders` o
on oi.order_id = o.order_id
join `target.customers` c
on o.customer_id = c.customer_id
group by c.customer_state)a
order by a.mean_freight_value
limit 5
```

e. Top 5 states with highest/lowest average time to delivery

Highest

```
select customer_state as state,
       round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))) as
highest_time_to_delivery
from `target.orders` o
join `target.customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by 2 desc
limit 5
```

| Row | state | highest_time_to_deliv |
|-----|-------|-----------------------|
| 1 | RR | 29.0 |
| 2 | AP | 27.0 |
| 3 | AM | 26.0 |
| 4 | AL | 24.0 |
| 5 | PA | 23.0 |

Lowest:

```
select customer_state as state,
       round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))) as
lowest_time_to_delivery
from `target.orders` o
join `target.customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by 2
limit 5
```

| Row | state | lowest_time_to_deliv |
|-----|-------|----------------------|
| 1 | SP | 8.0 |
| 2 | PR | 12.0 |
| 3 | MG | 12.0 |
| 4 | DF | 13.0 |
| 5 | SC | 14.0 |

- f. Top 5 states where delivery is really fast/ not so fast compared to estimated date
Fast delivery:

```
select customer_state as state,
       round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)))
as fast_delivery
from `target.orders` o
join `target.customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
```

```
group by c.customer_state
order by 2
limit 5
```

| Row | state | fast_delivery |
|-----|-------|---------------|
| 1 | AL | 8.0 |
| 2 | MA | 9.0 |
| 3 | SE | 9.0 |
| 4 | CE | 10.0 |
| 5 | MS | 10.0 |

The date between estimated delivery date and the day on which it is delivered in AL, MA, SE is least, hence we can say that these are the states where delivery is fast compared to estimated date

Not so fast:

```
select customer_state as state,
       round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)))
) as not_so_fast_delivery
from `target.orders` o
join `target.customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by 2 desc
limit 5
```

| Row | state | not_so_fast_delivery |
|-----|-------|----------------------|
| 1 | AC | 20.0 |
| 2 | AP | 19.0 |
| 3 | AM | 19.0 |
| 4 | RO | 19.0 |
| 5 | RR | 16.0 |

The date between estimated delivery date and the day on which it is delivered in states AC, AP, AM is most, hence we can say that these are the states where delivery is not so fast compared to estimated date.

6. Payment type analysis:

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

```
select a.payment_type,
       a.month,
       a.count_of_orders,
       lead(a.count_of_orders) over(partition by a.payment_type order by month) as
next_month,
       round((lead(a.count_of_orders) over(partition by a.payment_type order by
month) - a.count_of_orders)) as mom_count_of_orders
from
(select
  extract (month from order_purchase_timestamp) as month,
  payment_type,
  count (p.order_id) as count_of_orders

from `target.payments` p
join `target.orders` o
on p.order_id = o.order_id
group by 1,2
order by 1)a
```

| Row | payment_type | month | count_of_orders | next_month | mom_count_of_orders |
|-----|--------------|-------|-----------------|------------|---------------------|
| 1 | debit_card | 1 | 118 | 82 | -36.0 |
| 2 | debit_card | 2 | 82 | 109 | 27.0 |
| 3 | debit_card | 3 | 109 | 124 | 15.0 |
| 4 | debit_card | 4 | 124 | 81 | -43.0 |
| 5 | debit_card | 5 | 81 | 209 | 128.0 |
| 6 | debit_card | 6 | 209 | 264 | 55.0 |
| 7 | debit_card | 7 | 264 | 311 | 47.0 |
| 8 | debit_card | 8 | 311 | 43 | -268.0 |
| 9 | debit_card | 9 | 43 | 54 | 11.0 |
| 10 | debit_card | 10 | 54 | 70 | 16.0 |

From the above picture we can see the month over month orders for different payment types.

Here negative sign indicated the decrease in orders when compared to previous month.

b. Count of orders based on the no. of payment instalments

```
select payment_installments,
       count(order_id) as count_of_orders
from `target.payments`
group by 1
```

| Row | payment_installment | count_of_orders |
|-----|---------------------|-----------------|
| 1 | 0 | 2 |
| 2 | 1 | 52546 |
| 3 | 2 | 12413 |
| 4 | 3 | 10461 |
| 5 | 4 | 7098 |
| 6 | 5 | 5239 |
| 7 | 6 | 3920 |

From the above query we can say that most of the orders were taken for single instalment. This can be because of the interest rates provided by banks. Maybe if there is a provision of no cost emi, orders may take place with multiple no of payment instalments .

Recommendations

- Given the peak order periods at the end of the year in 2017 and 2018, consider developing special holiday marketing campaigns to capitalize on the increased demand.
- Allocate marketing resources more effectively by targeting promotions and advertisements during the afternoon and evening when customer activity is higher.
- Consider expanding distribution networks in areas with higher mean freight values to improve service and reduce shipping costs.
- Based on the observed increase in orders during specific periods, optimize inventory management to ensure that popular items are well-stocked during peak seasons, reducing the risk of stock outs and capitalizing on high demand.
- Since most orders are taken for single instalments, explore the feasibility of introducing a no-cost EMI option. This can attract customers who prefer spreading their payments over multiple instalments, potentially increasing order volume and customer satisfaction.

THANK YOU