

## ## 1. In Depth Exploration

**Q1. Data type of all columns in the "customers" table.**

```
select
  column_name, data_type
from `Target`.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers';
```

Row	column_name ▼	data_type ▼
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

**Q2. Get the time range between which the orders were placed.**

```
select
  min(order_purchase_timestamp) as First_order,
  max(order_purchase_timestamp) as Last_order
from `Target.orders`;
```

Row	First_order ▼	Last_order ▼
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

**Q3. Count the Cities & States of customers who ordered during the given period.**

```
select
  count(distinct c.customer_city) as City_Count,
  count(distinct c.customer_state) as State_Count
from `Target.customers` c
join `Target.orders` o
on c.customer_id = o.customer_id;
```

Row	City_Count ▼	State_Count ▼
1	4119	27

## ## 2. In-depth Exploration

Q1. Is there a growing trend in the no. of orders placed over the past years?

```
with cte1 as
(
  select
    distinct (year_month),
    count(order_id) as order_count
  from (
    select
      *,
      format_date('%Y-%m', order_purchase_timestamp) as year_month
    from `Target.orders`
  )
  group by 1
  order by 1)

select
  year_month,
  order_count,
  lag(order_count) over(order by year_month) as previous_month,
  concat(round(((order_count - lag(order_count) over(order by year_month)) / lag(order_count)
over(order by year_month))*100,2), " ", "%") as pct_change
from cte1
order by 1;
```

Row	year_month	order_count	previous_month	pct_change
1	2016-09	4	null	null
2	2016-10	324	4	8000 %
3	2016-12	1	324	-99.69 %
4	2017-01	800	1	79900 %
5	2017-02	1780	800	122.5 %
6	2017-03	2682	1780	50.67 %
7	2017-04	2404	2682	-10.37 %
8	2017-05	3700	2404	53.91 %

**Q2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?**

```
select
  distinct (year_month),
  count(order_id) as order_count
from (
  select
    *,
    format_date('%Y-%m',order_purchase_timestamp) as year_month
  from `Target.orders`
)
group by 1
order by 1;
```

Row	year_month	order_count
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700

**Q3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)**

```
-- 0-6 hrs : Dawn
-- 7-12 hrs : Mornings
-- 13-18 hrs : Afternoon
-- 19-23 hrs : Night
```

```
select
  case when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
  when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Mornings'
  when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
  else 'Night'
  end as time_of_day,
  count(order_id) as order_count
from `Target.orders`
group by 1
order by 2 desc;
```

Row	time_of_day	order_count
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

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

**Q1. Get the month on month no. of orders placed in each state.**

```
select
    c.customer_state,
    format_date('%Y-%m', o.order_purchase_timestamp) as year_month,
    format_date('%B', o.order_purchase_timestamp) AS month_name,
    count(o.order_id) as order_count
from `Target.customers` c
inner join `Target.orders` o
on c.customer_id = o.customer_id
group by 1,2,3
order by 1,2
```

Row	customer_state	year_month	month_name	order_count
1	AC	2017-01	January	2
2	AC	2017-02	February	3
3	AC	2017-03	March	2
4	AC	2017-04	April	5
5	AC	2017-05	May	8
6	AC	2017-06	June	4
7	AC	2017-07	July	5
8	AC	2017-08	August	4

**Q2. How are the customers distributed across all the states?**

```
select
    c.customer_state as State_Name,
    count(distinct o.order_id) as order_count
from `Target.customers` c
inner join `Target.orders` o
on c.customer_id = o.customer_id
group by 1
order by 2 desc;
```

Row	State_Name	order_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140

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

**Q1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).**

```
with cte1 as (
    select
        format_date('%Y',o.order_purchase_timestamp) as year,
        format_date('%m',o.order_purchase_timestamp) as month,
        round(sum(p.payment_value),2) as Total_Payment
    from `Target.orders` o
    left join `Target.payments` p
    on o.order_id = p.order_id
    group by 1,2
    order by 1 asc,2 asc
),

cte2 as (
    select
        month,
        sum(case when year = '2017' then Total_Payment else 0 end) as Total_Payment_2017,
        sum(case when year = '2018' then Total_Payment else 0 end) as Total_Payment_2018,
    from cte1
    where month between '01' and '08'
    group by 1
    order by 1
)

select
    month,
    Total_Payment_2017,
    Total_Payment_2018,
    round(((Total_Payment_2018 - Total_Payment_2017)/(Total_Payment_2017)) * 100,2) as Pct_change
from cte2;
```

Row	month	Total_Payment_2017	Total_Payment_2018	Pct_change
1	01	138488.04	1115004.18	705.13
2	02	291908.01	992463.34	239.99
3	03	449863.6	1159652.12	157.78
4	04	417788.03	1160785.48	177.84
5	05	592918.82	1153982.15	94.63
6	06	511276.38	1023880.5	100.26
7	07	592382.92	1066540.75	80.04
8	08	674396.32	1022425.32	51.61

**Q2. Calculate the Total & Average value of order price for each state.**

```
select
    c.customer_state,
    round(sum(p.payment_value),2) as Total_Payment,
    round(avg(p.payment_value),2) as Avg_Payment
from `Target.customers` c
left join `Target.orders` o
on c.customer_id = o.customer_id
left join `Target.payments` p
on o.order_id = p.order_id
group by 1
order by 1;
```

Row	customer_state	Total_Payment	Avg_Payment
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76

**Q3. Calculate the Total & Average value of order freight for each state.**

```
select
    c.customer_state,
    round(sum(o2.freight_value),2) as Total_Freight,
    round(avg(o2.freight_value),2) as Avg_Freight
from `Target.customers` c
left join `Target.orders` o
on c.customer_id = o.customer_id
left join `Target.order_items` o2
on o.order_id = o2.order_id
group by 1
order by 1;
```

Row	customer_state	Total_Freight	Avg_Freight
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77

## ## 5. Analysis based on sales, freight and delivery time.

Q1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

```
with cte as (  
select  
    order_purchase_timestamp as Order_date,  
    order_delivered_customer_date as Delivery_date,  
    order_estimated_delivery_date as Estimated_delivery_date,  
    date_diff(order_delivered_customer_date, order_purchase_timestamp, Day) as time_to_deliver,  
    date_diff(order_estimated_delivery_date, order_delivered_customer_date, Day) as  
diff_estimated_delivery  
from `Target.orders` o  
order by 4 asc, 5 asc  
)  
  
select  
    Order_date,  
    Delivery_date,  
    Estimated_delivery_date,  
    time_to_deliver,  
    diff_estimated_delivery  
from cte  
where time_to_deliver is not null
```

Row	Order_date	Delivery_date	Estimated_delivery_date	time_to_deliver	diff_estimated_delivery
1	2017-05-15 11:50:53 UTC	2017-05-16 10:21:52 UTC	2017-05-24 00:00:00 UTC	0	7
2	2018-06-18 12:59:42 UTC	2018-06-19 12:43:27 UTC	2018-06-28 00:00:00 UTC	0	8
3	2018-05-18 15:03:19 UTC	2018-05-19 12:28:30 UTC	2018-05-29 00:00:00 UTC	0	9
4	2018-05-14 12:20:06 UTC	2018-05-15 12:17:46 UTC	2018-05-25 00:00:00 UTC	0	9
5	2017-06-19 08:19:45 UTC	2017-06-19 21:07:52 UTC	2017-06-30 00:00:00 UTC	0	10
6	2017-11-16 13:54:08 UTC	2017-11-17 13:49:40 UTC	2017-11-29 00:00:00 UTC	0	11
7	2017-07-04 11:37:47 UTC	2017-07-05 08:09:26 UTC	2017-07-17 00:00:00 UTC	0	11
8	2017-05-31 12:00:35 UTC	2017-06-01 10:28:24 UTC	2017-06-13 00:00:00 UTC	0	11
9	2018-06-28 14:34:48 UTC	2018-06-29 14:12:18 UTC	2018-07-12 00:00:00 UTC	0	12

**Q2. Find out the top 5 states with the highest & lowest average freight value.**

```

with cte1 as (
    select
        c.customer_state,
        round(avg(o2.freight_value),2) as Avg_Freight,
        dense_rank() over(order by round(avg(o2.freight_value),2) desc) as rnk
    from `Target.customers` c
    left join `Target.orders` o
    on c.customer_id = o.customer_id
    left join `Target.order_items` o2
    on o.order_id = o2.order_id
    group by 1
),

top5 as (
    select
        customer_state,
        Avg_Freight,
        rnk
    from cte1
    where rnk <= 5
),

bottom5 as (
    select
        customer_state,
        Avg_Freight,
        rnk
    from cte1
    order by rnk desc
    limit 5
)

select
    'Top 5' AS Category,
    customer_state,
    Avg_Freight
from top5
union all
select
    'Bottom 5' AS Category,
    customer_state,
    Avg_Freight
from bottom5
order by Category desc, Avg_Freight desc;

```

Row	Category	customer_state	Avg_Freight
1	Top 5	RR	42.98
2	Top 5	PB	42.72
3	Top 5	RO	41.07
4	Top 5	AC	40.07
5	Top 5	PI	39.15
6	Bottom 5	DF	21.04
7	Bottom 5	RJ	20.96
8	Bottom 5	MG	20.63
9	Bottom 5	PR	20.53
10	Bottom 5	SP	15.15



**Q3. Find out the top 5 states with the highest & lowest average delivery time.**

```

with cte as (
    select
        c.customer_state,
        avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, Day)) as
time_to_delivery
    from `Target.orders` o
    right join `Target.customers` c
    on o.customer_id = c.customer_id
    where o.order_delivered_customer_date is not null
    group by 1
),

ranked_data as (
    select
        customer_state as state,
        time_to_delivery as avg_delivery_time,
        dense_rank() over(order by time_to_delivery asc) as rnk_asc,
        dense_rank() over(order by time_to_delivery desc) as rnk_desc
    from cte
)

select
    'Top 5' as category,
    state,
    avg_delivery_time
from ranked_data
where rnk_asc <= 5
union all
select
    'Bottom 5' as category,
    state,
    avg_delivery_time
from ranked_data
where rnk_desc <= 5
order by category, avg_delivery_time;

```

Row	category	state	avg_delivery_time
1	Bottom 5	PA	23.32
2	Bottom 5	AL	24.04
3	Bottom 5	AM	25.99
4	Bottom 5	AP	26.73
5	Bottom 5	RR	28.98
6	Top 5	SP	8.3
7	Top 5	PR	11.53
8	Top 5	MG	11.54
9	Top 5	DF	12.51
10	Top 5	SC	14.48

**Q4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**

```
select
    c.customer_state,
    round(avg(datetime_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date,
Day)),2) as diff_estimated_delivery
from `Target.customers` c
left join `Target.orders` o
on c.customer_id = o.customer_id
where o.order_delivered_customer_date is not null and order_estimated_delivery_date is not null
group by 1
order by 2 asc
limit 5
```

Row	customer_state	diff_estimated_delivery
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

**## 6. Analysis based on the payments:**

**Q1. Find the month on month no. of orders placed using different payment types.**

```
select
    format_date('%B-%Y',o.order_purchase_timestamp) as month_year,
    p.payment_type,
    count(o.order_id) as order_count
from `Target.orders` o
left join `Target.payments` p
on o.order_id = p.order_id
group by 1,2
order by 1 asc
```

Row	month_year	payment_type	order_count
1	April-2017	voucher	202
2	April-2017	credit_card	1846
3	April-2017	UPI	496
4	April-2017	debit_card	27
5	April-2018	credit_card	5455
6	April-2018	UPI	1287
7	April-2018	debit_card	97
8	April-2018	voucher	370

**Q2. Find the no. of orders placed on the basis of the payment installments that have been paid.**

```
select
  distinct(p.payment_installments),
  count(o.order_id) as order_count
from `Target.orders` o
left join `Target.payments` p
on o.order_id = p.order_id
where p.payment_installments is not null and p.payment_installments != 0
group by 1
order by 1 asc
```

Row	payment_installment	order_count
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268

**Insights:**

**1. Seasonal Payment Trends:**

- Certain months show a higher usage of specific payment methods, potentially linked to promotions or customer behavior.

**2. Installment Usage:**

- Installments are preferred during large purchases, suggesting customer sensitivity to upfront costs.

**3. Time Range of Orders:**

- The company's operations span a wide time frame, providing a substantial dataset for analyzing seasonal and yearly trends.

**4. Geographical Spread:**

- The orders come from diverse cities and states, indicating a good reach across the country.

**5. Delivery Efficiency:**

- Certain states have significantly faster delivery times compared to the estimated delivery date.

**6. Challenging States:**

- Some states have consistently high delivery times, possibly due to logistical challenges or remote locations.

**7. Top States for Fast Delivery:**

- Some states have consistently high delivery times, possibly due to logistical challenges or remote locations.

**8. Freight Costs:**

- Freight costs vary significantly across states, with some states showing much higher average freight charges.

**Recommendations:**

**1. Payment Campaigns:**

- Launch targeted campaigns during peak months, offering incentives for specific payment types.

**2. Upsell Opportunities:**

- Promote high-value products with attractive installment options to increase average order value.

**3. Expand Marketing Campaigns:**

- Focus marketing efforts during peak ordering times (e.g., afternoon and evening) to boost sales.

**4. Target Underrepresented States:**

- Identify and target states with low customer activity to improve market penetration.

**5. Improve Logistics:**

- Focus on states with slower delivery times by optimizing routes, increasing warehouse coverage, or collaborating with local courier services.

**6. Reward Fast Deliveries:**

- Promote fast delivery states as examples of company reliability in marketing campaigns.

**7. Leverage Delivery Success:**

- Highlight states with exceptional delivery performance as part of customer testimonials and case studies.

**8. Optimize Freight Costs:**

- Negotiate better shipping rates with logistics partners for states with high freight costs.