

TARGET CASE STUDIES

- 1.Initial Exploration

--o **Get the time range between which the orders were placed.**

```
select
    min(order_purchase_timestamp) as min_date,
    max(order_purchase_timestamp) as max_date,
    datediff(day , min(order_purchase_timestamp), max(order_purchase_timestamp)) as
time_range_between_orders
from orders
```

	min_date	max_date	time_range_between_orders
1	2016-09-04 21:15:19.0000000	2018-10-17 17:30:18.0000000	773

--o **Count the cities & states of customers who ordered during the given period.**

```
SELECT
    count(distinct customer_city) as City_count,
    count(distinct customer_state) as State_count
FROM
    customers as c JOIN orders as o
    ON c.customer_id = o.customer_id;
```

	City_count	State_count
1	4119	27

-- 2. In-depth Exploration

-- o Identify trends in the number of orders placed over the years.

```
select
    year(order_purchase_timestamp) as Years,
    datename(month,order_purchase_timestamp) as month_name,
    count(order_purchase_timestamp) as Number_of_Orders
from
    orders
group by
    year(order_purchase_timestamp),datename(month,order_purchase_timestamp)
order by
    year(order_purchase_timestamp),datename(month,order_purchase_timestamp)
```

	Years	month_name	Number_of_Orders
1	2016	December	1
2	2016	October	324
3	2016	September	4
4	2017	April	2404
5	2017	August	4331
6	2017	December	5673
7	2017	February	1780
8	2017	January	800
9	2017	July	4026
10	2017	June	3245
11	2017	March	2682

-- o Detect any monthly seasonality in the number of orders.

```
select
    month(order_purchase_timestamp) as Months,
    datename(month,order_purchase_timestamp) as month_name,
    count(*) as number_orders
from
    orders
group by
    month(order_purchase_timestamp),
    datename(month,order_purchase_timestamp)
order by month(order_purchase_timestamp),
    datename(month,order_purchase_timestamp)
```

	Months	month_name	number_orders
1	1	January	8069
2	2	February	8508
3	3	March	9893
4	4	April	9343
5	5	May	10573
6	6	June	9412
7	7	July	10318
8	8	August	10843
9	9	September	4305
10	10	October	4959
11	11	November	7544

-- o Determine the time of day when Brazilian customers mostly place orders (Dawn, Morning, Afternoon, or Night).

```

select
case when datepart(hour,order_purchase_timestamp) > 0 and
datepart(hour,order_purchase_timestamp) < 6 then 'Dawn'
when datepart(hour,order_purchase_timestamp) > 6 and
datepart(hour,order_purchase_timestamp) < 12 then 'Morning'
when datepart(hour,order_purchase_timestamp) > 12 and
datepart(hour,order_purchase_timestamp) < 18 then 'Afternoon'
else 'Night'
end as Duration,
count(*) as Total_count
from orders
group by
case when datepart(hour,order_purchase_timestamp) > 0 and
datepart(hour,order_purchase_timestamp) < 6 then 'Dawn'
when datepart(hour,order_purchase_timestamp) > 6 and
datepart(hour,order_purchase_timestamp) < 12 then 'Morning'
when datepart(hour,order_purchase_timestamp) > 12 and
datepart(hour,order_purchase_timestamp) < 18 then 'Afternoon'
else 'Night'
end
order by count(*) desc

```

	Duration	Total_count
1	Night	42991
2	Afternoon	32366
3	Morning	21738
4	Dawn	2346

-- 3. Evolution of E-commerce Orders in Brazil

--o Month-on-month number of orders placed in each state.

```
select
    c.customer_state,
    datetime(month,o.order_purchase_timestamp) as Months,
    count(o.order_purchase_timestamp) as Counts
from
    orders as o
join
    customers as c
on
    o.customer_id = c.customer_id
group by
    c.customer_state,
    datetime(month,o.order_purchase_timestamp)
order by c.customer_state
```

	customer_state	Months	Counts
1	AC	April	9
2	AC	August	7
3	AC	December	5
4	AC	February	6
5	AC	January	8
6	AC	July	9

--o Distribution of customers across all states.

```
select customer_state, count(customer_id) as number_of_customers
from customers
group by customer_state
order by count(customer_id) desc
```

	customer_state	number_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637

-- 4. Impact on Economy - Analyze money movement by looking at order prices, freight, and other factors.
 -- o Calculate the percentage increase in the cost of orders from 2017 to 2018 (Jan-Aug).

```
with cte1 as
(
select o.*,p.payment_value
from orders o
join payments p
on o.order_id = p.order_id
where year(o.order_purchase_timestamp) between 2017 and 2018 and
month(o.order_purchase_timestamp)between 1 and 8
),
cte2 as
(
select year(order_purchase_timestamp) as Years,
sum(payment_value) as cost
from cte1
group by year(order_purchase_timestamp)
)

select Years, round(cost,2) as cost,
round(lead(cost)over(order by Years),2),
(((lead(cost)over(order by Years)-cost)/cost)*100) as Percent_increase
from cte2
```

	Years	cost	(No column name)	Percent_increase
1	2017	3669022.12	8694733.84	136.976871791763
2	2018	8694733.84	NULL	NULL

-- o Calculate the total & average value of order prices for each state.

```
select c.customer_state,
sum(p.payment_value) as total_price,
AVG(p.payment_value) as average_price
from payments as p
join orders as o on p.order_id = o.order_id
join customers as c on c.customer_id = o.customer_id
group by c.customer_state
order by c.customer_state
```

	customer_state	total_price	average_price
1	AC	19680.6198749542	234.293093749455
2	AL	96962.0602003336	227.077424356753
3	AM	27966.9300133586	181.603441645186
4	AP	16262.8000450134	232.325714928763
5	BA	616645.820546836	170.816016771977
6	CE	279464.030406773	199.902739919008

-- o **Calculate the total & average value of freight prices for each state.**

```
select c.customer_state,
round(sum(oi.freight_value),2) as total_price_of_freights,
round(AVG(oi.freight_value),2) as average_price_of_freights
from order_items as oi
join orders as o on oi.order_id = o.order_id
join customers as c on c.customer_id = o.customer_id
group by c.customer_state
order by c.customer_state
```

	customer_state	total_price_of_freights	average_price_of_freights
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71

--5. Analysis on Sales, Freight, and Delivery Time

--o **Calculate the delivery time and the difference between estimated and actual delivery dates.**

```
select
datediff(day,order_purchase_timestamp,order_estimated_delivery_date) as
estimated_days,
datediff(day,order_purchase_timestamp,order_delivered_customer_date) as
actual_delivery_time,
datediff(day,order_delivered_customer_date,order_estimated_delivery_date) as diff
from orders
where order_status = 'delivered'
```

	estimated_days	actual_delivery_time	diff
1	16	8	8
2	20	14	6
3	27	9	18
4	27	14	13
5	13	3	10
6	23	17	6

--o Identify the top 5 states with the highest & lowest average freight values.

```
select top 5 c.customer_state, avg(oi.freight_value) as AVG_top_5_freight_value
from order_items oi
join orders o
on oi.order_id = o.order_id
join customers c
on c.customer_id = o.customer_id
group by c.customer_state
order by AVG_top_5_freight_value desc
```

	customer_state	AVG_top_5_freight_value
1	RR	42.9844230505136
2	PB	42.7238040353571
3	RO	41.0697122752238
4	AC	40.0733695548514
5	PI	39.1479705145438

```
select top 5 c.customer_state, avg(oi.freight_value) as AVG_lowest_5_freight_value
from order_items as oi join orders as o
on oi.order_id = o.order_id join customers as c on c.customer_id = o.customer_id
group by c.customer_state
order by AVG_lowest_5_freight_value
```

	customer_state	AVG_lowest_5_freight_value
1	SP	15.1472754114787
2	PR	20.5316516108019
3	MG	20.6301668497907
4	RJ	20.960923974036
5	DF	21.041354959161

--o Identify the top 5 states with the highest & lowest average delivery times.

```
select top 5 c.customer_state,
avg(datediff(day, o.order_purchase_timestamp, o.order_delivered_customer_date)) as
AVG_highest_5_delivery_times
from orders as o
join customers as c
on o.customer_id = c.customer_id
group by c.customer_state
order by AVG_highest_5_delivery_times desc
```

	customer_state	AVG_highest_5_delivery_times
1	RR	29
2	AP	27
3	AM	26
4	AL	24
5	PA	23

```

select top 5 c.customer_state,
avg(datediff(day,o.order_purchase_timestamp,o.order_delivered_customer_date)) as
AVG_lowest_5_delivery_times
from orders as o
join customers as c
on o.customer_id = c.customer_id
group by c.customer_state
order by AVG_lowest_5_delivery_times

```

	customer_state	AVG_lowest_5_delivery_times
1	SP	8
2	PR	11
3	MG	11
4	DF	12
5	SC	14

--o **Identify the top 5 states where delivery is faster than the estimated date.**

```

SELECT top 5
    c.customer_state,
    DATEDIFF(DAY, o.order_estimated_delivery_date, o.order_delivered_customer_date) AS
faster_delivery_days
FROM orders AS o JOIN customers AS c
ON o.customer_id = c.customer_id
WHERE o.order_status = 'delivered'
AND o.order_delivered_customer_date < o.order_estimated_delivery_date
#-- Ensure delivery is faster
ORDER BY
faster_delivery_days
# -- Sort by the fastest deliveries (smallest negative values)

```

	customer_state	faster_delivery_days
1	SP	-147
2	MA	-140
3	RS	-135
4	SP	-124
5	RJ	-109

--6. Analysis Based on Payments

--o Month-on-month number of orders placed using different payment types.

```
select payment_type,MONTH(order_purchase_timestamp) as months,  
year(order_purchase_timestamp) as years,  
  
datename(month,order_purchase_timestamp) as month_name, count(payment_type) as  
number_of_orders  
from orders as o  
join payments as p on o.order_id = p.order_id  
group by year(order_purchase_timestamp),  
MONTH(order_purchase_timestamp),datename(month,order_purchase_timestamp),  
payment_type
```

	payment_type	months	years	month_name	number_of_orders
1	credit_card	9	2016	September	3
2	credit_card	10	2016	October	254
3	debit_card	10	2016	October	2
4	UPI	10	2016	October	63
5	voucher	10	2016	October	23
6	credit_card	12	2016	December	1

--o Number of orders based on payment installments.

```
select payment_installments, count(payment_installments) as number_of_orders  
from payments  
where payment_installments > 1  
group by payment_installments  
order by count(payment_installments) desc
```

	payment_installments	number_of_orders
1	2	12413
2	3	10461
3	4	7098
4	10	5328
5	5	5239
6	8	4268

Insights

- *The data covers a time period of about 2 years.*
- *Between September 2016 and November 2017, the number of orders generally increased, showing overall growth. However, this growth slowed down in December 2017, then bounced back in January 2018, remained stable for a while, and then sharply declined again in September 2018.*
- *Orders tend to be higher in the middle months of the year (May, June, July, August), likely because of summer holiday shopping, suggesting these months are popular for shopping.*
- *September has the lowest order count in this dataset. It could be helpful to look into possible reasons for the sudden and large drop in orders during this month.*
- *The majority of orders were placed in the night, followed by the afternoon.*
- *The cost of orders increased by 136.98% from 2017 to 2018.*
- *Out of all orders, approx 87,000 were delivered before the estimated delivery date.*
- *The most commonly used payment method was credit card, followed by UPI.*