

TARGET SQL PROJECT

Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analysing the given dataset to extract valuable insights and provide actionable recommendations.

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

a) Data type of all columns in the "customers" table.

```
SELECT column_name, data_type
FROM `target-sql-390205.SQL_Project.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
```

Query results		
JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH		
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

b) Get the time range between which the orders were placed.

```
SELECT MIN(order_purchase_timestamp) AS FIRST_ORDER_TS,
MAX(order_purchase_timestamp) AS LAST_ORDER_TS ,
MIN(DATE(order_purchase_timestamp)) AS FIRST_ORDER_DATE,
MAX(DATE(order_purchase_timestamp)) AS LAST_ORDER_DATE,
FROM `SQL_Project.orders`
```

Query results					
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	FIRST_ORDER_TS ▼	LAST_ORDER_TS ▼	FIRST_ORDER_DATE	LAST_ORDER_DATE	
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	2016-09-04	2018-10-17	

- From this we could say that the dataset contains data between September 2016 to October 2018

c) Count the number of Cities and States in our dataset.

```
SELECT count(distinct geolocation_city) as City_Count,
count(distinct geolocation_state) as State_Count,
from `SQL_Project.geolocation`
```

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTI
Row	City_Count ▼	State_Count ▼		
1	8011	27		

- There are 8011 cities and 27 states in our dataset.

Q2: In-depth Exploration

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

```
select extract(year from order_purchase_timestamp) as year,
count(order_id) as NumberOfOrders
from `SQL_Project.orders`
group by extract(year from order_purchase_timestamp)
```

Query results			
JOB INFORMATION		RESULTS	JSON
		EXECUTION DETAILS	
Row	Year	NumberOfOrders	
1	2016	329	
2	2017	45101	
3	2018	54011	

- We could clearly see that there is a growing trend over years from this Query.

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

```
select
format_timestamp('%B',order_purchase_timestamp)as Month,
count(order_id) as NumberOfOrders
from `SQL_Project.orders`
group by 1
order by 2 desc
```

Query results			
JOB INFORMATION		RESULTS	JSON
		EXECUTION DETAILS	
Row	Month	NumberOfOrders	
1	August	10843	
2	May	10573	
3	July	10318	
4	March	9893	
5	June	9412	
6	April	9343	
7	February	8508	
8	January	8069	
9	November	7544	
10	December	5674	
11	October	4850	

- From the output we can say August,May & July are the Top 3 months in terms of orders placed over the given set of Years.

**c) During what time of the day, do the Brazilian customers mostly place their orders?
(Dawn, Morning, Afternoon or 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 'Morning'  
  
when extract(HOUR from order_purchase_timestamp) between 13 and 18 then 'Afternoon'  
  
else 'Night'  
  
end as TimeOfDay,  
  
Count(order_id) as NumberOfOrders  
  
from `SQL_Project.orders`  
  
group by 1  
  
order by 2 desc
```

Query results			
JOB INFORMATION		RESULTS	JSON
		EXECUTION DETAILS	
Row	TimeOfDay ▼	NumberOfOrders ▼	
1	Afternoon	38135	
2	Night	28331	
3	Morning	27733	
4	Dawn	5242	

- From this we can conclude that Most of the Brazilian customers prefer to place order on Afternoon time and The Least preferred is Dawn time of the day.

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

a) Get the month on month no. of orders placed in each state.

```
select  
  
format_timestamp('%B',o.order_purchase_timestamp)as Month,  
  
C.customer_state as State,  
  
count(o.order_id) as NumberOfOrders  
  
from `SQL_Project.orders` O  
  
join `SQL_Project.customers` C on O.customer_id = C.customer_id  
  
group by 1,2  
  
order by 3 desc
```

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
EXECUTION GRAPH				
Row	Month	State	NumberOfOrders	
1	August	SP	4982	
2	May	SP	4632	
3	July	SP	4381	
4	June	SP	4104	
5	March	SP	4047	
6	April	SP	3967	
7	February	SP	3357	
8	January	SP	3351	
9	November	SP	3012	
10	December	SP	2357	

- We get month over month order count for each state and from the results it's evident that SP (san paulo) is the top state in terms of orders placed.

b) How are the customers distributed across all the states?

```
select  
  
count(distinct customer_id) as CustomerCount,  
  
customer_state  
  
from `SQL_Project.customers`  
  
group by 2  
  
order by 1 desc
```

Query results		
JOB INFORMATION		
RESULTS		
JSON		
EXECUTION DETAILS		
Row	CustomerCount ▼	customer_state ▼
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO

- Most of the Customers are from SP,RJ,MG states.

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

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

```
with calculations as (select

sum(case when extract(YEAR from o.order_purchase_timestamp)=2017

and extract(month from o.order_purchase_timestamp) between 1 and 8 then
p.payment_value

else 0

end)as TOTAL2017,

sum(case when extract(YEAR from o.order_purchase_timestamp)=2018

and extract(month from o.order_purchase_timestamp) between 1 and 8 then
p.payment_value

else 0

end)as TOTAL2018,

from `SQL_Project.payments` p

join `SQL_Project.orders` o on p.order_id = o.order_id)

select ROUND(((TOTAL2018-TOTAL2017) / TOTAL2017)*100,2) AS PercentageIncreased

from calculations
```

Query results

JOB INFORMATION		RESULTS
Row	PercentageIncreased	
1	136.98	

- From 2017 to 2018 the cost of orders had increased by 136.98% (considering Jan-Aug).

b) Calculate the Total & Average value of order price for each state.

```
select  
  
c.customer_state as State,  
  
Round(sum(ot.price),2) as TotalPrice,  
  
Round(avg(ot.price),2) as AvgPrice  
  
from `SQL_Project.order_items` ot  
  
join `SQL_Project.orders` o on ot.order_id = o.order_id  
  
join `SQL_Project.customers` c on o.customer_id = c.customer_id  
  
group by 1  
  
order by 2 desc,3 desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	State ▼	TotalPrice ▼	AvgPrice ▼	
1	SP	5202955.05	109.65	
2	RJ	1824092.67	125.12	
3	MG	1585308.03	120.75	
4	RS	750304.02	120.34	
5	PR	683083.76	119.0	
6	SC	520553.34	124.65	
7	BA	511349.99	134.6	
8	DF	302603.94	125.77	
9	GO	294591.95	126.27	
10	ES	275037.31	121.91	

- From the State by State result we can see that SP,RJ,MG are the states that are leading the charts.

c) Calculate the Total & Average value of order freight for each state.

```
select
c.customer_state as State,
Round(sum(ot.freight_value),2) as TotalFreight,
Round(avg(ot.freight_value),2) as AvgFreight

from `SQL_Project.order_items` ot
join `SQL_Project.orders` o on ot.order_id = o.order_id
join `SQL_Project.customers` c on o.customer_id = c.customer_id

group by 1
order by 2 desc,3 desc
```

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	State ▼	TotalFreight ▼	AvgFreight ▼	
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	
10	DF	50625.5	21.04	

- From the State by State result we can see that SP,RJ,MG are the states that have higher Freight charges on whole.

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

a) 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.

```
select
order_id,
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 `SQL_Project.orders`
ORDER BY 1
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	
Row	order_id ▼	time_to_deliver ▼	diff_estimated_delive		
1	00010242fe8c5a6d1ba2dd792...	7	8		
2	00018f77f2f0320c557190d7a1...	16	2		
3	000229ec398224ef6ca0657da...	7	13		
4	00024acbcd0a6daa1e931b03...	6	5		
5	00042b26cf59d7ce69dfabb4e...	25	15		
6	00048cc3ae777c65dbb7d2a06...	6	14		
7	00054e8431b9d7675808bcb8...	8	16		
8	000576fe39319847cbb9d288c...	5	15		
9	0005a1a1728c9d785b8e2b08b...	9	0		
10	0005f50442cb953dcd1d21e1f...	2	18		

b) Find out the top 5 states with the highest & lowest average freight value.

Top 5 states with the highest average freight value:

```
select
c.customer_state as State, Round(avg(ot.freight_value),2) as AvgFreight
from `SQL_Project.order_items` ot
join `SQL_Project.orders` o on ot.order_id = o.order_id
join `SQL_Project.customers` c on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DE
Row	State ▼	AvgFreight ▼		
1	RR	42.98		
2	PB	42.72		
3	RO	41.07		
4	AC	40.07		
5	PI	39.15		

Top 5 states with the lowest average freight value:

```
select
c.customer_state as State,
Round(avg(ot.freight_value),2) as AvgFreight
from `SQL_Project.order_items` ot
join `SQL_Project.orders` o on ot.order_id = o.order_id
join `SQL_Project.customers` c on o.customer_id = c.customer_id
group by 1
order by 2
limit 5
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUT
Row	State ▼	AvgFreight ▼		
1	SP	15.15		
2	PR	20.53		
3	MG	20.63		
4	RJ	20.96		
5	DF	21.04		

c)Find out the top 5 states with the highest & lowest average delivery time.

Top 5 states with the Lowest average delivery time.

```
select
c.customer_state,
ROUND(avg(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY)
),2) as AvgDeliveryTime
from `SQL_Project.customers` C
join `SQL_Project.orders` O on O.customer_id = C.customer_id
group by 1
order by 2
limit 5
```

Query results		
JOB INFORMATION		
RESULTS		
JSON		
EXECUTED		
Row	customer_state	AvgDeliveryTime
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

Top 5 states with the Highest average delivery time.

```
select
c.customer_state,
ROUND(avg(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY)
),2) as AvgDeliveryTime
from `SQL_Project.customers` C
join `SQL_Project.orders` O on O.customer_id = C.customer_id
group by 1
order by 2 desc
limit 5
```

Query results		
JOB INFORMATION		
RESULTS		
JSON		
EXECUTED		
Row	customer_state	AvgDeliveryTime
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

d) 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(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,
DAY)),2) AS AvgQuickDelivery

from `SQL_Project.customers` C
join `SQL_Project.orders` O on O.customer_id = C.customer_id

group by 1
order by 2 desc
limit 5
```

Query results

JOB INFORMATION		RESULTS	JSON	EXEC
Row	customer_state ▼	AvgQuickDelivery ▼		
1	AC	19.76		
2	RO	19.13		
3	AP	18.73		
4	AM	18.61		
5	RR	16.41		

- These are all the Top 5 States where orders are being delivered faster than expected.

6)Analysis based on the payments:

a)Find the month on month no. of orders placed using different payment types.

```
select
format_timestamp('%B',o.order_purchase_timestamp)as Month,
p.payment_type,
count(o.order_id) as TotalOrders
from `SQL_Project.orders` o
join `SQL_Project.payments` p on o.order_id = p.order_id
group by 1,2
order by 3 desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	Month ▼	payment_type ▼	TotalOrders ▼		
1	May	credit_card	8350		
2	August	credit_card	8269		
3	July	credit_card	7841		
4	March	credit_card	7707		
5	April	credit_card	7301		
6	June	credit_card	7276		
7	February	credit_card	6609		
8	January	credit_card	6103		
9	November	credit_card	5897		
10	December	credit_card	4378		

- From the results, we found out that Credit Card is the most preferred payment type followed by UPI and Voucher.

b)Find the no. of orders placed on the basis of the payment instalments that have been paid.

```
SELECT payment_installments, COUNT(DISTINCT order_id) AS Total_orders
FROM `SQL_Project.payments`
GROUP BY 1
order by 2 desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	payment_installment	Total_orders ▼	
1	1	49060	
2	2	12389	
3	3	10443	
4	4	7088	
5	10	5315	
6	5	5234	
7	8	4253	
8	6	3916	
9	7	1623	
10	9	644	

- This query helps us to get the count of orders placed by different Instalment options.
- Most of the orders were paid in single payment.

In case if we need only the Count of Orders that have been completed full payment, then we can use the following query:

```
SELECT payment_installments, count(distinct order_id) as Full_Paid_Count
FROM `SQL_Project.payments`
where payment_installments = payment_sequential
group by 1
order by 1
```

JOB INFORMATION		RESULTS	JSON
Row	payment_installment	Full_Paid_Count ▼	
1	1	48236	
2	2	53	
3	3	1	

7) Actionable Insights & Recommendations:

- Given that August, May, and July are the top three months in terms of orders placed, allocate additional marketing resources and promotional activities during these months to maximise sales opportunities.
- Since most Brazilian customers prefer to place orders in the afternoon, ensure that your website or app is optimised for a seamless and user-friendly shopping experience during this time.
- As SP, RJ, and MG are the states with the highest number of customers, focus on building strong customer relationships in these regions. Implement personalised marketing strategies, loyalty programs to retain existing customers and attract new ones.
- Since RR, PB, and RO have higher freight charges, explore ways to optimise shipping costs for customers in these states. Negotiate partnerships with logistics providers to secure better shipping rates.
- Simplify Payment Process for Single Payments: Since most orders were paid in a single payment, ensure a seamless and straightforward checkout process for customers opting for this payment method.