Business Case: Target SQL

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

1. Data type of columns in a table

QUERY

```
SELECT column_name, data_type

FROM `scaler-dsml-sql-378217.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

2. Time period for which the data is given

In the orders table, we can find the time period using the below **query**. From the -

```
SELECT      min(DATE(order_purchase_timestamp)) AS min_date,
max(DATE(order_purchase_timestamp)) as max_date
FROM       `Target.orders` ]
```

Row	min_date	max_date
1	2016-09-04	2018-10-17

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

QUERY

```
SELECT     customer_city, customer_state
FROM       `Target.orders` o
JOIN       `Target.customers` c     on  c.customer_id = o.customer_id
GROUP BY   customer_city, customer_state
```

Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT

2. In-depth Exploration:

1. 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?

TREND of ecommerce can be seen by the increase/decrease of either the number of orders or the amount of sales. Since payments (sales value) for specific period of 2017 and 2018 will be compared later on, we are relating trends with the number of orders as that also closely indicates the Ecommerce activity of people involved.

QUERY

```
With query_1 as

(select count(distinct order_id) as Number_of_orders , format_datetime("%B",order_purchase_timestamp) as month, extract(year from order_purchase_timestamp) as year, extract(month from order_purchase_timestamp) as month_number

from Target.orders

group by month, year, month_number

order by year, month_number),


Trend as

(select Number_of_orders , month, year,month_number, sum(Number_of_orders) over (order by year, month_number rows between unbounded preceding and current row) AS Monthly_Running_total,sum(Number_of_orders) over (order by year, month_number rows between 1 preceding and 1 following) AS Three_Month_Running_total, Round((Number_of_orders - LAG(Number_of_orders, 1) OVER (ORDER BY year, month_number))*100 / LAG(Number_of_orders, 1) OVER (ORDER BY year, month_number),2) AS monthly_percent_increase

from query_1

order by year, month_number),
```

```
Seasonality as

(select Number_of_orders, month , year , Monthly_Running_total  , Three_Month_Running_total ,
CONCAT(monthly_percent_increase,' %') as monthly_pct_increase , CASE

    WHEN SUM(CASE WHEN monthly_percent_increase > 0 THEN 1 ELSE 0 END) OVER (

      ORDER BY year,month_number

      ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING

    ) >= 2 THEN 'Yes'

    ELSE 'No'

  END AS atleast_2_positive_change_in_3_rows

from Trend

order by   year, CASE month

             WHEN 'January' THEN 1

             WHEN 'February' THEN 2

             WHEN 'March' THEN 3

             WHEN 'April' THEN 4

             WHEN 'May' THEN 5

             WHEN 'June' THEN 6

             WHEN 'July' THEN 7

             WHEN 'August' THEN 8

             WHEN 'September' THEN 9

             WHEN 'October' THEN 10

             WHEN 'November' THEN 11

             WHEN 'December' THEN 12

         END )
```

```
select *

from Seasonality
```

Row	Number_of_orders	month	year	Monthly_Running_total	Three_Month_Running_total	monthly_pct_increase	atleast_2_positive_change_in_3_rows
1	4	September	2016	4	328	*null*	No
2	324	October	2016	328	329	8000 %	No
3	1	December	2016	329	1125	-99.69 %	Yes
4	800	January	2017	1129	2581	79900 %	Yes
5	1780	February	2017	2909	5262	122.5 %	Yes
6	2682	March	2017	5591	6866	50.67 %	Yes
7	2404	April	2017	7995	8786	-10.37 %	Yes
8	3700	May	2017	11695	9349	53.91 %	No
9	3245	June	2017	14940	10971	-12.3 %	Yes
10	4026	July	2017	18966	11602	24.07 %	Yes

Insights and recommendations-

We can observe a general upward trend in the monthly number of orders for the year 2017. However, when we examine the trend on a month-to-month basis, we notice that it fluctuates considerably. In order to assess seasonality, we looked for instances where there were at least two consecutive months showing a positive change in the number of orders, with one preceding month, the current month, and one following month in the stack. We observed such instances in the months of February, July, and August.

My recommendation would be to investigate whether the increase in orders during these three months is linked to specific holidays or festivals. If so, it would indicate that consumers are more inclined to shop during festive periods, which could help E-commerce companies develop strategies to capitalize on these occasions. Additionally, offering benefits such as vouchers and other perks could be an effective way to attract consumers.

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

Time has been divided as follows-

Dawn- 5 am to 9 am

Morning- 9 am to 12 am

Afternoon- 12 am to 5 pm

Night - 5 pm to 5 am

QUERY

```
select count(order_id)as number_of_orders, X.time_of_day
from ( select  order_id  , hour_of_day,
     case when hour_of_day in (5,6,7,8) then "Dawn"
               when hour_of_day in (9,10,11) then 'Morning'
               when hour_of_day in (12,13,14,15,16) then 'Afternoon'
               else "Night"
               end as time_of_day
       from (select distinct order_id,
EXTRACT(HOUR FROM order_purchase_timestamp) AS hour_of_day
from `Target.orders`))X
group by X.time_of_day
```

Row	number_of_orde	time_of_day
1	17540	Morning
2	44802	Night
3	32211	Afternoon
4	4888	Dawn

Insights and recommendations-

It is evident that the majority of E-commerce transactions in Brazil occur during the night and afternoon hours. This trend can be attributed to individuals preparing for work in the morning and completing early morning tasks. The morning period is typically hectic for most people, and they may not have the time or inclination to engage in E-commerce activities. However, during the afternoon lunch period, people may have more free time to engage in transactions. Furthermore, after office hours, which usually end around 5 P.M., individuals have more leisure time, and this is when most E-commerce transactions occur.

To encourage more transactions, E-commerce platforms should display offers for consumers during the night and afternoon hours when activity is highest. Additionally, customer support services should be made available during these hours to ensure smooth and uninterrupted E-commerce activities and transactions..

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

Get month on month orders by states

To accurately show the month on month orders by states in the Brazil region, it would be best to include both the month and the year in the output. This will provide a complete picture of how the orders are evolving over time.

QUERY

```
select     count(order_id) as Number_of_orders, c.customer_state , month_name, year_number

from

(SELECT      distinct order_id, customer_id ,   format_datetime("%B",order_purchase_timestamp)
as month_name ,   extract(year from order_purchase_timestamp) as year_number

from Target.orders)x

join `Target.customers`c     on   c.customer_id = x.customer_id

group by   c.customer_state,month_name, year_number

order by   c.customer_state,   year_number, CASE month_name

              WHEN 'January' THEN 1

              WHEN 'February' THEN 2

              WHEN 'March' THEN 3

              WHEN 'April' THEN 4

              WHEN 'May' THEN 5

              WHEN 'June' THEN 6

              WHEN 'July' THEN 7

              WHEN 'August' THEN 8

              WHEN 'September' THEN 9
```

```
        WHEN 'October' THEN 10
        WHEN 'November' THEN 11
        WHEN 'December' THEN 12
    END
```

Row	Number_of_orders	customer_state	month_name	year_number
1	2	AC	January	2017
2	3	AC	February	2017
3	2	AC	March	2017
4	5	AC	April	2017
5	8	AC	May	2017
6	4	AC	June	2017
7	5	AC	July	2017
8	4	AC	August	2017
9	5	AC	September	2017
10	6	AC	October	2017

Distribution of customers across the states in Brazil

QUERY

```
SELECT  count(customer_unique_id) as Total_customers, customer_state
from `Target.customers`
```

```
group by customer_state

order by count(customer_unique_id) desc
```

Row	Total_customers	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

Insights and recommendations-

This dataset contains 27 rows, representing the 27 states in Brazil. Notably, states such as Sao Paulo, Rio de Janeiro, and Minas Gerais have the highest number of orders.

To improve sales in states with lower order numbers, I recommend leveraging local climate and festivals to better market products to consumers. Additionally, we should analyze and implement successful strategies used by the top-performing states in these areas. Increasing marketing and advertising efforts in these regions, as well as exploring the purchasing power of consumers in these states, could also lead to increased sales.

4. 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

Query-

```
With Query_1 as

(SELECT

  format_datetime("%B", O.order_purchase_timestamp) as month_name,

    round(SUM(CASE  WHEN  extract(year  from  O.order_purchase_timestamp)  =  2017  THEN
T.payment_value ELSE 0 END),2) as payment_in_2017,

    round(SUM(CASE  WHEN  extract(year  from  O.order_purchase_timestamp)  =  2018  THEN
T.payment_value ELSE 0 END),2) as payment_in_2018

FROM `Target.payments` T

JOIN `Target.orders` O ON T.order_id = O.order_id

WHERE extract(year from O.order_purchase_timestamp) in (2017, 2018) and extract(month
from O.order_purchase_timestamp) between 1 and 8

GROUP BY   month_name),



Query_2 as

(

SELECT                  month_name,         payment_in_2017,         payment_in_2018,
CONCAT(round((((payment_in_2018)-(payment_in_2017))*100/payment_in_2017),2),  "  %")  as
change_in_payment_2017_to_2018

from Query_1

order by CASE month_name

            WHEN 'January' THEN 1
```

```
            WHEN 'February' THEN 2
            WHEN 'March' THEN 3
            WHEN 'April' THEN 4
            WHEN 'May' THEN 5
            WHEN 'June' THEN 6
            WHEN 'July' THEN 7
            WHEN 'August' THEN 8
        END)
SELECT
  month_name,
  payment_in_2017,
  payment_in_2018,
  change_in_payment_2017_to_2018
FROM Query_2
UNION ALL
SELECT
  'Total',
  ROUND(SUM(payment_in_2017), 2),
  ROUND(SUM(payment_in_2018), 2),
    CONCAT(ROUND((((SUM(payment_in_2018)) - (SUM(payment_in_2017))) * 100 /
SUM(payment_in_2017)), 2), " %")
FROM Query_1
ORDER BY CASE month_name
            WHEN 'January' THEN 1
            WHEN 'February' THEN 2
            WHEN 'March' THEN 3
```

```
        WHEN 'April' THEN 4
        WHEN 'May' THEN 5
        WHEN 'June' THEN 6
        WHEN 'July' THEN 7
        WHEN 'August' THEN 8
        ELSE 9
    END
```

Row	month_name	payment_in_2017	payment_in_2018	change_in_payment_2017_to_2018
1	January	138488.04	1115004.18	705.13 %
2	February	291908.01	992463.34	239.99 %
3	March	449863.6	1159652.12	157.78 %
4	April	417788.03	1160785.48	177.84 %
5	May	592918.82	1153982.15	94.63 %
6	June	511276.38	1023880.5	100.26 %
7	July	592382.92	1066540.75	80.04 %
8	August	674396.32	1022425.32	51.61 %
9	Total	3669022.12	8694733.84	136.98 %

Insights and recommendations-

Based on the data analysis, it is evident that the total payment in 2018 has experienced a significant increase compared to 2017, with a growth rate of 136.98%. This suggests a surge in E-commerce activity in the given period. However, it is noticeable that the payments have remained stagnant in 2018 during the given period.

A potential recommendation to enhance E-commerce activity during the same months of the upcoming years would be to investigate the reason behind the continuous growth observed in 2017 and apply the findings. Improvements could be made in various areas, such as platform functionality, payment convenience, enhanced marketing and advertising strategies, and the introduction of incentives, such as coupons and vouchers, for using E-commerce platforms. These measures could potentially improve E-commerce activity and drive growth in the future.

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

Query

```
select  round(sum(freight_value),2) as sum_freight_value , round(avg(freight_value),2) as mean_freight_value , round(sum(price),2) as sum_price_value, round(avg(price),2) as mean_price_value, customer_state

from `Target.order_items` I

join  `Target.orders` O     on  I.order_id = O.order_id

join `Target.customers` C    on  O.customer_id = C.customer_id

group by customer_state

order by customer_state
```

Row	sum_freight_value	mean_freight_value	sum_price_value	mean_price_value	customer_state
1	3686.75	40.07	15982.95	173.73	AC
2	15914.59	35.84	80314.81	180.89	AL
3	5478.89	33.21	22356.84	135.5	AM
4	2788.5	34.01	13474.3	164.32	AP
5	100156.68	26.36	511349.99	134.6	BA
6	48351.59	32.71	227254.71	153.76	CE
7	50625.5	21.04	302603.94	125.77	DF
8	49764.6	22.06	275037.31	121.91	ES
9	53114.98	22.77	294591.95	126.27	GO
10	31523.77	38.26	119648.22	145.2	MA

5. Analysis on sales, freight and delivery time

Calculate days between purchasing, delivering and estimated delivery

Query-

```
select          date_diff(order_delivered_customer_date  ,order_purchase_timestamp,  day)  as time_to_delivery,

              date_diff(order_estimated_delivery_date ,order_delivered_customer_date, day) as diff_estimated_delivery


from `Target.orders`

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL
```

Row	time_to_delivery	diff_estimated_delivery
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	29	1
7	43	-4
8	40	-4
9	37	-1
10	33	-5

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

Top 5 states with highest average freight value

Query-

```
select  customer_state, round(avg(freight_value),2) as mean_freight_value ,

        round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp, day)),1)
as  average_time_to_delivery,

        round(avg(date_diff(order_estimated_delivery_date ,order_delivered_customer_date,
day)),1) as average_diff_estimated_delivery

from `Target.order_items` I

join  `Target.orders` O      on   I.order_id = O.order_id

join `Target.customers` C    on   O.customer_id = C.customer_id

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL

group by customer_state

order by mean_freight_value desc

limit 5
```

Row	customer_state	mean_freight_value	average_time_to_delivery	average_diff_estimated_delivery
1	PB	43.09	20.1	12.2
2	RR	43.09	27.8	17.4
3	RO	41.33	19.3	19.1
4	AC	40.05	20.3	20.0
5	PI	39.12	18.9	10.7

Top 5 states with lowest average freight value

Query-

```
select  customer_state, round(avg(freight_value),2) as mean_freight_value ,

        round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp, day)),1)
as  average_time_to_delivery,

           round(avg(date_diff(order_estimated_delivery_date  ,order_delivered_customer_date,
day)),1) as average_diff_estimated_delivery

from `Target.order_items` I

join  `Target.orders` O      on   I.order_id = O.order_id

join `Target.customers` C    on   O.customer_id = C.customer_id

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL

group by customer_state

order by mean_freight_value

limit 5
```

Row	customer_state	mean_freight_value	average_time_to_delivery	average_diff_estimated_delivery
1	SP	15.11	8.3	10.3
2	PR	20.47	11.5	12.5
3	MG	20.63	11.5	12.4
4	RJ	20.91	14.7	11.1
5	DF	21.07	12.5	11.3

Top 5 states with Highest average time to delivery

Query-

```
select  customer_state, round(avg(freight_value),2) as mean_freight_value ,

        round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp, day)),1)
as  average_time_to_delivery,

           round(avg(date_diff(order_estimated_delivery_date  ,order_delivered_customer_date,
day)),1) as average_diff_estimated_delivery

from `Target.order_items` I

join  `Target.orders` O      on   I.order_id = O.order_id

join `Target.customers` C    on   O.customer_id = C.customer_id

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL

group by customer_state

order by average_time_to_delivery desc

limit 5
```

Row	customer_state	mean_freight_value	average_time_to_delivery	average_diff_estimated_delivery
1	AP	34.16	27.8	17.4
2	RR	43.09	27.8	17.4
3	AM	33.31	26.0	19.0
4	AL	35.87	24.0	8.0
5	PA	35.63	23.3	13.4

Top 5 states with Lowest average time to delivery

Query-

```
select  customer_state, round(avg(freight_value),2) as mean_freight_value ,

        round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp, day)),1)
as  average_time_to_delivery,

           round(avg(date_diff(order_estimated_delivery_date  ,order_delivered_customer_date,
day)),1) as average_diff_estimated_delivery

from `Target.order_items` I

join  `Target.orders` O      on   I.order_id = O.order_id

join `Target.customers` C    on   O.customer_id = C.customer_id

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL

group by customer_state

order by average_time_to_delivery

limit 5
```

Row	customer_state	mean_freight_value	average_time_to_delivery	average_diff_est
1	SP	15.11	8.3	10.3
2	MG	20.63	11.5	12.4
3	PR	20.47	11.5	12.5
4	DF	21.07	12.5	11.3
5	SC	21.51	14.5	10.7

Top 5 states where delivery is really fast compared to estimated date

Query-

```
select  customer_state, round(avg(freight_value),2) as mean_freight_value ,

        round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp, day)),1)
as  average_time_to_delivery,

           round(avg(date_diff(order_estimated_delivery_date  ,order_delivered_customer_date,
day)),1) as average_diff_estimated_delivery

from `Target.order_items` I

join  `Target.orders` O      on   I.order_id = O.order_id

join `Target.customers` C    on   O.customer_id = C.customer_id

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL

group by customer_state

order by average_diff_estimated_delivery desc

limit 5
```

Row	customer_state	mean_freight_value	average_time_to_delivery	average_diff_estimated_delivery
1	AC	40.05	20.3	20.0
2	RO	41.33	19.3	19.1
3	AM	33.31	26.0	19.0
4	AP	34.16	27.8	17.4
5	RR	43.09	27.8	17.4

Top 5 states where delivery is not so fast compared to estimated date

Query-

```
select  customer_state, round(avg(freight_value),2) as mean_freight_value ,

        round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp, day)),1)
as  average_time_to_delivery,

           round(avg(date_diff(order_estimated_delivery_date  ,order_delivered_customer_date,
day)),1) as average_diff_estimated_delivery

from `Target.order_items` I

join  `Target.orders` O      on   I.order_id = O.order_id

join `Target.customers` C    on   O.customer_id = C.customer_id

where order_estimated_delivery_date is not null and order_delivered_customer_date is not NULL

group by customer_state

order by average_diff_estimated_delivery

limit 5
```

Row	customer_state	mean_freight_value	average_time_to_delivery	average_diff_estimated_delivery
1	AL	35.87	24.0	8.0
2	MA	38.49	21.2	9.1
3	SE	36.57	21.0	9.2
4	ES	22.03	15.2	9.8
5	BA	26.49	18.8	10.1

Insights and recommendations-

In the Brazilian ecommerce market, there appears to be a correlation between freight costs and delivery times. States with lower average freight costs tend to have faster delivery times, while states with faster delivery times than estimated may have higher freight costs. This could be due to more efficient logistics systems and investments in infrastructure in states with lower freight costs, while logistical challenges and distance from transportation hubs could contribute to longer delivery times and higher costs in states with higher freight costs. However, there may be other factors at play and regional variations that would require further analysis.

My recommendations would be -

- Investment in logistics infrastructure i.e. transportation systems and warehouses to reduce delivery times and lower costs.
- Negotiate better rates with carriers.
- Leverage technology such as route optimization software and real-time tracking to improve delivery efficiency and reduce freight costs.
- Streamlining operations such as supply chain and warehouse operations
- Improving customer communication for accurate and timely information about delivery times to help manage expectations

6. Payment type analysis:

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

QUERY-

```
select    format_datetime("%B", O.order_purchase_timestamp) as month_name, count(P.order_id)
as count_of_orders , P.payment_type, extract(year from O.order_purchase_timestamp) as year

from    `Target.payments` P

join    `Target.orders` O         on      P.order_id = O.order_id

group by payment_type, year, month_name

order by year , CASE month_name

              WHEN 'January' THEN 1

             WHEN 'February' THEN 2

             WHEN 'March' THEN 3

             WHEN 'April' THEN 4

             WHEN 'May' THEN 5

             WHEN 'June' THEN 6

             WHEN 'July' THEN 7

             WHEN 'August' THEN 8

             WHEN 'September' THEN 9

             WHEN 'October' THEN 10

             WHEN 'November' THEN 11

             WHEN 'December' THEN 12

         END
```

Row	month_name	count_of_orders	payment_type	year
1	September	3	credit_card	2016
2	October	254	credit_card	2016
3	October	23	voucher	2016
4	October	2	debit_card	2016
5	October	63	UPI	2016
6	December	1	credit_card	2016
7	January	61	voucher	2017
8	January	197	UPI	2017
9	January	583	credit_card	2017
10	January	9	debit_card	2017

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

Query-

```
select count(order_id) as count_of_orders, payment_installments

from  `Target.payments`

group by payment_installments
```

Row	count_of_orders	payment_installments
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8
10	644	9

Insights and recommendations-

As the payment instalments of orders are increasing, the count of orders is decreasing and it significantly reduces if the instalments period is over an year . The trend above indicates that consumers are becoming more cautious with their spending and are opting for more affordable payment options. This trend may be due to the economic situation in Brazil, where many consumers are facing financial challenges and may be less willing to take on debt. Also,

This trend requires ecommerce companies to re-evaluate pricing strategies and payment options to cater to the changing needs of consumers. Lower interest rates can attract and retain customers in this changing landscape.