Business Case 1: Target SQL

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

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

Screenshot :

Field name	Type	Mode
customer_id	STRING	NULLABLE
customer_unique_id	STRING	NULLABLE
customer_zip_code_prefix	INTEGER	NULLABLE
customer_city	STRING	NULLABLE
customer_state	STRING	NULLABLE

Insights:

We can find the data types of all the columns which are present in the customers table using “Information_schema.columns”.

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

Query :

```
select
min(order_purchase_timestamp) as Start_date,
max(order_purchase_timestamp) as end_date
from target.orders;
```

Screenshot:

Row	Start_date	end_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insight:

Among all the orders present in the orders table the start date is “2016-09-04” and the End date is “2018-10-17”.

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

Query :

```
select
c.customer_city,
c.customer_state,
count(*) as total_count
from target.customers as c inner join target.orders o
on c.customer_id=o.customer_id
where o.order_purchase_timestamp
between (select min(order_purchase_timestamp) as Start_date from target.orders) and
(select max(order_purchase_timestamp) as end_date from target.orders)
group by 1,2
order by total_count desc
```

Screenshot :

Row	customer_city	customer_state	total_count
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521
6	campinas	SP	1444
7	porto alegre	RS	1379
8	salvador	BA	1245
9	guarulhos	SP	1189
10	sao bernardo do campo	SP	938
11	niteroi	RJ	849

Insight :

The highest orders were placed in the city named ‘sao paulo’ from state SP and followed by the ‘rio de janerio’ from state RJ and others.

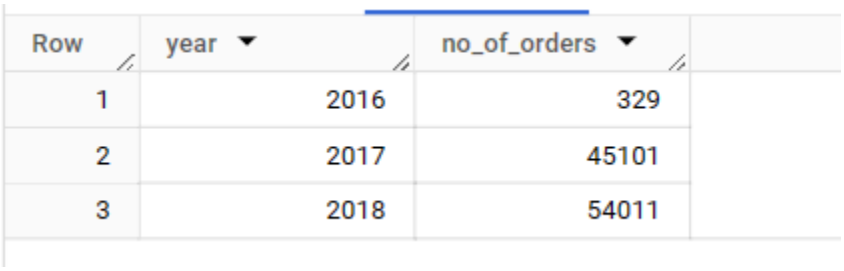
❖ In-depth Exploration:

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

Query :

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

Screenshot :



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

Insights:

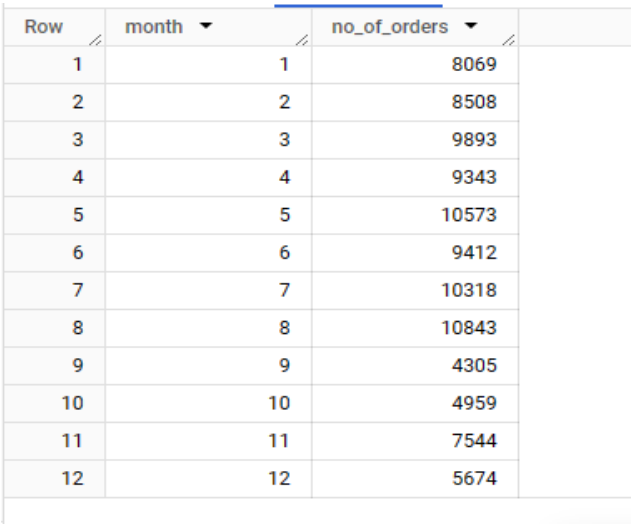
The data indicates a substantial increase in the number of orders, showing an impressive surge of 13,608% from 2016 to 2017. Following this, there was a more moderate growth of 19% from 2017 to 2018.

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

Query :

```
select extract(month from order_purchase_timestamp) as month,
count(*) as no_of_orders
from `target.orders`
group by extract(month from order_purchase_timestamp)
order by month
```

Screenshot :



Row	month	no_of_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

Insight:

The trend in the number of orders exhibited an increase until August, following which there was a decline. Months such as May, July, and August saw a surge in order volume, whereas September recorded the lowest number of orders placed.

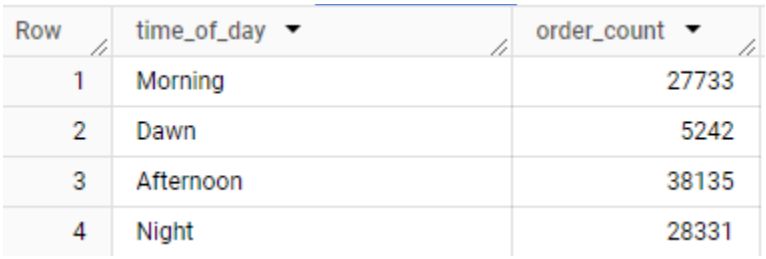
6. **During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)**
    - 0-6 hrs : Dawn
    - 7-12 hrs : Mornings
    - 13-18 hrs : Afternoon
    - 19-23 hrs : Night

Query :

```
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'
when extract (hour from order_purchase_timestamp) between 19 AND 23 then 'Night'
end as time_of_day,
count(*) as order_count from `target.orders`
```

group by time_of_day;

Screenshot :



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

Insights:
Brazilian customers primarily place their orders during the afternoon, with morning and night closely following in frequency but dawn experiences a significantly decreased number of orders compared to other times of the day.

Recommendations:
Nevertheless, there is a substantial volume of orders placed during both the morning and night periods. To increase the order count, we can provide customers with additional products that complement their original purchase or provide additional discounts to increase the order count.

❖ Evolution of E-commerce orders in the Brazil region:

7. Get the month-on-month no. of orders placed in each state.

Query :

```
select
distinct
extract(month from o.order_purchase_timestamp) as month,
c.customer_state,
count(*) over(partition by c.customer_state,extract(month from
o.order_purchase_timestamp)) as no_of_orders
from target.orders o join target.customers c
on o.customer_id=c.customer_id
order by 1,2
```

Screenshot :

Row	customer_state	month	order_count
1	AC	1	8
2	AL	1	39
3	AM	1	12
4	AP	1	11
5	BA	1	264
6	CE	1	99
7	DF	1	151
8	ES	1	159
9	GO	1	164
10	MA	1	66
11	MG	1	971

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

Query :

```
select distinct
customer_state,
count(*) over(partition by customer_state) as state_wise_customer_count
from target.customers
order by 2 desc
```

Screenshot :

Row	customer_state	state_wise_customer_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
9	ES	2033
10	GO	2020

Insight:
The data reveals a significant concentration of customers in the state of SP, with RJ trailing closely behind, and a diverse distribution across various other states.

Recommedation :
There are lowest number of customers from AP, AC and RR states, Target should open stores or run exclusive offers for those geographics locations.

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

9. **Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).**
You can use the "payment_value" column in the payments table to get the cost of orders.

Query :

```
with a as (select
extract(year from o.order_purchase_timestamp) as Year,
sum(p.payment_value) as Total_cost_over_year,
lag(sum(p.payment_value)) over(order by sum(p.payment_value)) as Previous_cost
from target.orders o join target.payments p
on o.order_id=p.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 year
order by 1)
select Year,Total_cost_over_year,Previous_cost,concat(round(((total_cost_over_year-previous_cost)/previous_cost)*100,2),'%') as Percentage_increase from a

Screenshot :

Row	Year	Total_cost_over_year	Previous_cost	Percentage_increase
1	2017	3669022.119999...	*null*	*null*
2	2018	8694733.839999...	3669022.119999...	136.98%

Insights:
There was a notable 136.98% increase observed from 2017 to 2018.

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

Query :

```
select
distinct
c.customer_state,
sum(p.payment_value) over w as Total_revenue,
round(avg(p.payment_value) over w,2) as Average_revenue
from target.customers c
join  target.orders o
on c.customer_id=o.customer_id
join target.payments p
on p.order_id=o.order_id
window w as(partition by c.customer_state)
order by Total_revenue desc,Average_revenue
```

Screenshot:

Row	customer_state	Total_revenue	Average_revenue
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
10	ES	325967.55	154.71

Insights:

1. Average revenue per customer is highest in PB, AC and RO states.
2. States SP, RJ and MG generate highest revenue for target among all other states.

Recommendation:

1. Target should focus on states (SP, PR, MG) where average per customer revenue is lower.
2. Since SP and MG has one of the highest revenue and lower average revenue means revenue per customer is lower, so it should run offers for bulk orders.

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

Query :

```
select
distinct
c.customer_state,
sum(ot.freight_value) over w as Total_fright_value,
round(avg(ot.freight_value) over w,2) as Avg_fright_value
from target.customers c
join  target.orders o
on c.customer_id=o.customer_id
join target.order_items ot
```

on o.order_id=ot.order_id
window w as(partition by c.customer_state)
order by Total_fright_value desc

Screenshot:

Row	customer_state	Total_fright_value	Avg_fright_value
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	117851.6800000...	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.65999999...	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

Insights:

1. Average fright value per customer is highest in RR, PB and RO states.
2. The states of SP, RJ, and MG have the top three highest freight values among the 27 states

❖ Analysis based on sales, freight and delivery time.

12. **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.
You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
    - **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
    - **diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date

Query :

```
select
order_purchase_timestamp,
order_estimated_delivery_date,
order_delivered_customer_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day)as
diff_estimated_delivery
from target.orders
order by time_to_deliver desc
```

Screenshot:

:

Row	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	time_to_deliver	diff_estimated_deliv
1	2017-02-21 23:31:27 UTC	2017-03-22 00:00:00 UTC	2017-09-19 14:36:39 UTC	209	181
2	2018-02-23 14:57:35 UTC	2018-03-15 00:00:00 UTC	2018-09-19 23:24:07 UTC	208	188
3	2017-03-07 23:59:51 UTC	2017-04-07 00:00:00 UTC	2017-09-19 15:12:50 UTC	195	165
4	2017-03-09 13:26:57 UTC	2017-04-11 00:00:00 UTC	2017-09-19 14:38:21 UTC	194	161
5	2017-03-08 22:47:40 UTC	2017-04-06 00:00:00 UTC	2017-09-19 14:00:04 UTC	194	166
6	2017-03-08 18:09:02 UTC	2017-04-17 00:00:00 UTC	2017-09-19 14:33:17 UTC	194	155
7	2018-01-03 09:44:01 UTC	2018-01-19 00:00:00 UTC	2018-07-13 20:51:31 UTC	191	175
8	2017-03-13 20:17:10 UTC	2017-04-05 00:00:00 UTC	2017-09-19 17:00:07 UTC	189	167
9	2017-03-15 11:24:27 UTC	2017-04-13 00:00:00 UTC	2017-09-19 14:38:18 UTC	188	159
10	2017-03-16 11:36:00 UTC	2017-04-28 00:00:00 UTC	2017-09-19 16:28:58 UTC	187	144

Insight:
The actual delivery time varies for most orders, with the longest delivery period being 209 days (about 7 months), which is considerably lengthy.

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

Query :

```
with highest as(select
distinct
c.customer_state as Highest_customer_state,
row_number() over(order by avg(ot.freight_value) desc) as row_num,
round(avg(ot.freight_value),2)as average_value_of_Highest_customer_state
from target.customers c
join  target.orders o
on c.customer_id=o.customer_id
```

```
join target.order_items ot
on o.order_id=ot.order_id
group by c.customer_state
order by average_value_of_Highest_customer_state desc
limit 5),
lowest as (
  select
distinct
c.customer_state as Lowest_customer_state,
row_number() over(order by avg(ot.freight_value)) as row_num,
round(avg(ot.freight_value),2) as average_value_of_Lowest_customer_state
from target.customers c
join  target.orders o
on c.customer_id=o.customer_id
join target.order_items ot
on o.order_id=ot.order_id
group by c.customer_state
order by average_value_of_Lowest_customer_state
limit 5
)
select
h.Highest_customer_state,h.average_value_of_Highest_customer_state,l.Lowest_
customer_state,l.average_value_of_Lowest_customer_state from highest h join
lowest l on h.row_num=l.row_num
```

Screenshot:

Row	Highest_customer_state	average_value_of_Highest_customer_state	Lowest_customer_state	average_value_of_Lowest_customer_state
1	RR	42.98	SP	15.15
2	PB	42.72	PR	20.53
3	RO	41.07	MG	20.63
4	AC	40.07	RJ	20.96
5	PI	39.15	DF	21.04

Insights:

1. Based on the dataset, the top 5 states with the highest average freight values are RR, PB, RO, AC, and PI.

2. Based on the dataset, the top 5 states with the lowest average freight values are SP, PR, MG, RJ, and DF.

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

Query:

```
with lowest as(select
c.customer_state as Lowest_customer_state,
```

```
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,d
ay)),2) as avg_days_Lowest_customer_state,
row_number() over(order by
avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)))
as r
from target.orders o join target.customers c on o.customer_id=c.customer_id
group by c.customer_state
order by avg_days_Lowest_customer_state
limit 5),
highest as(
  select
c.customer_state as Highest_customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,d
ay)),2)as avg_days_Highest_customer_state,
row_number() over(order by
avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))
desc)  as r
from target.orders o join target.customers c on o.customer_id=c.customer_id
group by c.customer_state
order by avg_days_Highest_customer_state desc
limit 5
)
select
h.Highest_customer_state,h.avg_days_Highest_customer_state,l.Lowest_custome
r_state,l.avg_days_Lowest_customer_state from lowest l join highest h on l.r=h.r
```

Screenshot:

Row	Highest_customer_state	avg_days_Highest_customer_state	Lowest_customer_state	avg_days_Lowest_customer_state
1	RR	28.98	SP	8.3
2	AP	26.73	PR	11.53
3	AM	25.99	MG	11.54
4	AL	24.04	DF	12.51
5	PA	23.32	SC	14.48

Insight:

1. Based on the dataset, the top 5 states with the highest average delivery time taken are RR, AP, AM, AL, and PA are falling under the range (23-29).
2. Based on the dataset, the top 5 states with the lowest average delivery time taken are SP, PR, MG, DF, and SC.

3. The state of São Paulo (SP) demonstrates notably fast delivery times, as indicated by the average delivery time across orders.

15. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Query :

```
SELECT
customer_state,
round(AVG(date_diff(order_delivered_customer_date,
order_purchase_timestamp, day)) -
AVG(date_diff(order_delivered_customer_date,order_estimated_delivery_date,da
y)),2) as avg_days
FROM  `target.orders` AS a
JOIN  `target.customers` AS b
ON  a.customer_id = b.customer_id
GROUP BY customer_state
ORDER BY avg_days ASC
LIMIT 5
```

Screenshot:

Row	customer_state	avg_days
1	SP	18.43
2	DF	23.63
3	MG	23.84
4	PR	23.89
5	ES	24.95

Insight:

1.The state SP demonstrates notably fast delivery times, as indicated by the average delivery time across orders.

2. The state AP is experiencing extended delivery durations, as evidenced by the average delivery time.

❖ Analysis based on the payments:

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

Query :

```
select extract(month from ord.order_purchase_timestamp) as month,
sum(case when pmt.payment_type = 'UPI' then 1 else 0 end) as UPI,
sum(case when pmt.payment_type = 'credit_card' then 1 else 0 end) as
credit_card,
sum(case when pmt.payment_type = 'voucher' then 1 else 0 end) as voucher,
sum(case when pmt.payment_type = 'debit_card' then 1 else 0 end) as debit_card,
sum(case when pmt.payment_type = 'not_defined' then 1 else 0 end) as
not_defined
from target.orders as ord
join target.payments as pmt
on pmt.order_id = ord.order_id
group by month
order by month ;
```

Screenshot:

Row	month	UPI	credit_card	voucher	debit_card	not_defined
1	1	1715	6103	477	118	0
2	2	1723	6609	424	82	0
3	3	1942	7707	591	109	0
4	4	1783	7301	572	124	0
5	5	2035	8350	613	81	0
6	6	1807	7276	563	209	0
7	7	2074	7841	645	264	0
8	8	2077	8269	589	311	2
9	9	903	3286	302	43	1
10	10	1056	3778	318	54	0

Insight:

Analysis of the payment types reveals that the majority of orders were made using credit cards , followed by UPI compared to other payment methods.

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

Query :

```
select
payment_installments,
count(*) as no_of_order
from target.payments p join target.orders o on p.order_id=o.order_id
where payment_installments>=1
group by payment_installments
order by payment_installments
```

Screenshot:

Row	payment_installment	no_of_order
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

Insights:

Most people preferred one time payment for order , followed by 2 installments and 3 installments.

Insights from the whole SQL Analysis (Target - Brazil)

● **Customer Data:** The customer data contains various columns with information about customers who have made purchases from the target. Most columns are text-based, and some may be empty or NULL for certain customers.

● **Order History:** The first order was placed in September 2016, and the most recent one occurred in October 2018, with a span of 772 days in between.

● **Top Ordering Regions:** Sao Paulo had the highest order count, followed by Rio de Janeiro, Belo Horizonte, and other areas.

● **Order Growth:** There is a noticeable increase in the number of orders, starting from 329 in 2016 and reaching 54,000 by 2018.

● **Seasonal Trends:** Order numbers tend to rise during the second and third quarters of the year but drop unexpectedly in the fourth quarter. The peak months for orders were May, July, and August, while September, October, and December saw the lowest order counts.

● **Order Timing:** Brazilian customers appear to prefer placing orders during the afternoon and fewer orders during the dawn hours.

● C**ustomer Distribution:** The largest customer base is in the state of São Paulo, followed by Rio de Janeiro, Minas Gerais, Rio Grande do Sul, and others.

● **Order Value Uptick:** There was a 16.62% increase in order value observed between 2017 and 2018 (up to August).

● **Order Value Analysis:** The analysis provides insights into both total order value and average order value across various states, with notable states including SP, RJ, MG, and others.

● **Freight Value Analysis:** Similarly, the analysis offers insights into total freight value and average order freight value across states, with notable states again being SP, RJ, MG, and others.

● **Delivery Time Deviations:** Some orders experienced significant deviations in actual delivery times compared to the estimated times provided by Target, with deviations of more than 150 days in some cases.

● **Freight Value by State:** States like RR, PB, RO, AC, and PI exhibit higher average freight values (in the range of 39 to 43), while states like SP, PR, MG, RJ, and DF have lower average freight values (in the range of 15 to 21).

● **Delivery Time by State:** States like RR, AP, AM, AL, and PA have longer average delivery times (greater than 23 days), whereas states like SP, PR, MG, DF, and SC have shorter delivery times (around 8 to 15 days).

● **Actual vs. Expected Delivery:** Generally, the actual average delivery time tends to be shorter than the expected delivery time, especially in states such as RR, AP, AC, AM, and RO.

● **Payment Methods:** The majority of payments are made using credit cards, followed by UPI, vouchers, and debit cards.

● **Payment Installments:** The analysis highlights the relationship between the number of orders placed and the number of payment installments that have been paid since the purchase