BUSINESS CASE: TARGET SQL

Note: I have saved the Target Case Project as 'sql_project' in my BIG QUERY

1. **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.**

Sol.

```
SELECT  column_name, data_type
FROM
first-project-405815.sql_project.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

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

Insights: Data Type Discrepancy: All columns are STRING, except "customer_zip_cpde_prefix," which is INT64.

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

Sol.

```
SELECT MIN(order_purchase_timestamp) AS start_date ,
MAX(order_purchase_timestamp) AS end_date
FROM
`sql_project.orders`
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

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

Order Timeline Insight: Target witnessed a 2-year span of orders placed between "2016-09-04 21:15:19 UTC" and "2018-10-17 17:30:18 UTC."
Recommendations: N/A

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

Sol.

```
SELECT COUNT(DISTINCT customer_city) AS No_of_cities,
count(DISTINCT customer_state) AS No_of_states
FROM
(SELECT o.order_id,c.customer_id,c.customer_city,c.customer_state
FROM `sql_project.orders` AS o
JOIN `sql_project.customers` AS c
ON c.customer_id = o.customer_id) as t
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	No_of_cities	No_of_states
1	4119	27

Insights: Geographic Reach: Target draws orders from 27 states and 4,119 cities, showcasing extensive customer engagement.
Recommendations: Strategic Growth Potential: Expanding services to additional cities could yield a substantial increase in orders.

2. In-depth Exploration

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

Sol.

```
SELECT COUNT(order_id) AS No_of_orders,
EXTRACT(YEAR FROM order_purchase_timestamp) AS Year
FROM `sql_project.orders`
GROUP BY Year
ORDER BY Year
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	No_of_orders	Year
1	329	2016
2	45101	2017
3	54011	2018

Insights: Order Evolution Insight: Noticeable increase in orders over the years, with a rapid surge in the second year, and a consistent, albeit minimal, uptick in the third year. Initial lower volumes in 2016 suggest robust growth attributed to stellar overall performance.
Recommendations:

1. Targeted Promotion: Focus promotions and offers on states with the lowest sales, optimizing impact for the upcoming year.

2. Review-driven Enhancement: Monitor regional customer reviews to identify departmental shortcomings (sales, delivery, etc.) and take prompt corrective actions.
3. Memorable Branding: Craft a captivating and trendy tagline for lasting brand recall. Strategically leverage YouTube ads during sale periods for sustained consumer awareness.

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

Sol.

```
SELECT EXTRACT(month FROM order_purchase_timestamp) AS month ,
COUNT(*) AS no_of_orders
FROM `sql_project.orders`
    GROUP BY month
    ORDER BY month
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

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

Insights: Order Seasonality Insight: May, July, and August emerge as seasoned months with higher orders, potentially linked to festivals. Conversely, unseasoned months like September, October, and December exhibit lower activity. Addressing potential reasons, such as financial constraints or competitor performance, can optimize strategies for sustained growth.

Recommendations:

Seasonal Stock Optimization: Align clothing and essentials with seasonal changes to meet customer demand effectively.

Festive Theme Integration: Tailor ads, website themes, and promotions to align with festivals or holidays, enhancing customer engagement.

Strategic Clearance Sales: Conduct clearance sales every 4 months with discounts for old stock management.

Targeted Rewards: Identify high-order customers in specific months; reward them with coupons or vouchers for subsequent months with lower order volumes.

Retention through Reminders: Boost customer retention by sending reminders or memories of past purchases to those who made purchases last month, encouraging repeat visits.

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

Sol.

```
SELECT time_of_day,COUNT(*) as No_of_orders
FROM (
SELECT EXTRACT(hour FROM order_purchase_timestamp),
CASE
WHEN EXTRACT(hour FROM order_purchase_timestamp)
BETWEEN 0 AND 6 THEN "Dawm"
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"
WHEN EXTRACT(hour FROM order_purchase_timestamp)
BETWEEN 19 AND 23 THEN "Night"
END AS time_of_day
FROM `sql_project.orders`) AS t
GROUP BY time_of_day
ORDER BY No_of_orders DESC
LIMIT 1
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	time_of_day	No_of_orders
1	Afternoon	38135

Insights: Prime Shopping Hours: Customer preferences indicate a strong inclination towards shopping in the afternoon, with a notable decline in dawn hours. Optimize strategies to capture the peak afternoon shopping trend.

Recommendations:

All night Sale Innovation: Launch a midnight to 6 am All night Sale, offering discounts with a slightly increased minimum purchase amount for eligibility.

Pro Member Boost: Elevate discounts by 5% during dawn for Pro Members, encouraging regular customers to join the Pro Membership program.

Engaging Lucky Draw: Introduce a thrilling afternoon-to-dawn lucky draw, unveiling winners at midnight or dawn. Presence during the reveal triggers special gifts, driving traffic and potential sales through effective recommendation strategies.

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

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

Sol.

```
SELECT EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
COUNT(o.order_id) AS No_of_orders,c.customer_state FROM `sql_project.orders` AS o
JOIN `sql_project.customers` AS c
ON c.customer_id = o.customer_id
GROUP BY month,customer_state
```

ORDER BY customer_state,month

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	month	No_of_orders	customer_state
1	1	8	AC
2	2	6	AC
3	3	4	AC
4	4	9	AC
5	5	10	AC
6	6	7	AC
7	7	9	AC
8	8	7	AC
9	9	5	AC
10	10	6	AC
11	11	5	AC
12	12	5	AC

Insights: Order Disparity Insight: States like "SP" boast over 2000 monthly orders, while "RJ" sees around 1000. In contrast, some states record less than 100 monthly orders, possibly linked to lower average salaries and various socioeconomic factors. Addressing these disparities can enhance overall market penetration and customer engagement.
Recommendations: Enhanced Delivery Solutions: Address delivery issues by bolstering delivery staff, prioritizing coverage in every state to compete with local supermarkets. Regional Sales Strategy: Tailor product offerings for specific states, enhancing regional sales by stocking products likely to appeal to local preferences. Strategic Stock Utilization: Integrate old stock with orders from states with lower demand, offering promotions like one-plus-one or free gifts to incentivize purchases and clear inventory.

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

Sol.

```
SELECT customer_state,COUNT(DISTINCT customer_id) AS customers_count
FROM `sql_project.customers`
GROUP BY customer_state
ORDER BY customers_count DESC
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW

Row	customer_state	customers_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
11	PE	1652
12	CE	1336

Insights: “SP” state has more number of customers and “RR” has less number of customers
Recommendations:
Personalized Recommendations: Understand and leverage buying patterns in specific states to enhance recommendations and cater to individual customer preferences.
Exclusive Early Access: Offer customers in states with fewer orders early access during sales, ensuring product availability and fostering repeat visits.
Amplify Word of Mouth: Elevate overall service across departments, coupled with compelling discounts, to stimulate positive word-of-mouth publicity and attract a larger customer base compared to competitors.

4. **Impact on Economy: Analyze 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).**
   You can use the "payment_value" column in the payments table to get the cost of orders.

   Sol.

```
WITH table1 AS
(SELECT EXTRACT(year FROM o.order_purchase_timestamp) AS year ,
extract(month from o.order_purchase_timestamp) AS monthly,
p.payment_value FROM `sql_project.payments` AS p
JOIN `sql_project.orders` AS o
USING (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
ORDER BY year,monthly,payment_value),

table2 AS
(SELECT year, ROUND(SUM(payment_value)) AS total FROM table1
GROUP BY year
ORDER BY year)

SELECT '2017' AS from_year, '2018' AS to_year,
 ROUND( ( MAX(total) -MIN(total) ) / MIN(total)*100 ) AS percentage_increase
FROM table2
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS

Row	from_year	to_year	percentage_increase
1	2017	2018	137.0

Insights: Cost Surge Insight: Noteworthy 137% increase in order costs from 2017 to 2018, prompting further analysis into the contributing factors.
Recommendations:
Seasonal Clearance Impact: Clearance sales at the end or mid-year may contribute to a slight increase in the cost of orders.
Feedback Analysis Imperative: Thoroughly analyze customer feedback to identify areas of improvement and enhance overall order efficiency.

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

Sol.

```
WITH customer_order_details AS
(SELECT c.customer_id,c.customer_state,o.order_id,oi.price
FROM `sql_project.customers` AS c
JOIN `sql_project.orders` AS o
ON o.customer_id = c.customer_id
JOIN `sql_project.order_items` AS oi
ON o.order_id = oi.order_id)

SELECT customer_state,
ROUND(SUM(price),2) AS Total,ROUND(AVG(price),2) AS Avg
FROM customer_order_details
GROUP BY customer_state
ORDER BY total DESC, Avg
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	customer_state	Total	Avg
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
11	PE	262788.03	145.51
12	CE	227254.71	153.76

Insights: Top Spenders: Customers from the SP state outshine all others, demonstrating higher spending compared to customers from other states.

Recommendations:

Personalized Recommendations: Understand and leverage buying patterns in specific states to enhance recommendations and cater to individual customer preferences.

Exclusive Early Access: Offer customers in states with fewer orders early access during sales, ensuring product availability and fostering repeat visits.

Amplify Word of Mouth: Elevate overall service across departments, coupled with compelling discounts, to stimulate positive word-of-mouth publicity and attract a larger customer base compared to competitors.

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

Sol.

```
WITH customer_order_details AS
(SELECT c.customer_id,c.customer_state,o.order_id,oi.freight_value
```

```
FROM `sql_project.customers` AS c
JOIN `sql_project.orders` AS o
ON o.customer_id = c.customer_id
JOIN `sql_project.order_items` AS oi
ON o.order_id = oi.order_id)

SELECT customer_state,
ROUND(SUM(freight_value),2) AS Total,ROUND(AVG(freight_value),2) AS Avg
FROM customer_order_details
GROUP BY customer_state
ORDER BY total DESC, Avg
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	customer_state	Total	Avg
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
11	ES	49764.6	22.06
12	CE	48351.59	32.71

Insights: Freight Analysis Insight: High order frequency from SP state contributes to elevated freight values, potentially influenced by higher taxes, distant warehouses, or delivery hubs. Contrastingly, RR state exhibits lower freight, calculated as the average of the sum divided by the number of orders placed from that city.

Recommendations:

Strategic Infrastructure: Establishing warehouses or delivery hubs in high-order states can optimize transportation costs and enhance operational efficiency. Tax-Adjusted Pricing: Customize product prices based on respective state taxes to offer competitive and regionally relevant pricing for increased customer satisfaction and market competitiveness.

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.
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:

i. **time_to_deliver = order_delivered_customer_date - order_purchase_timestamp**
ii. **diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date**

Sol.

```
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 time_to_deliver DESC
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	order_id	time_to_deliver	diff_estimated_deliv
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	0f4519c5f1c541ddec9f21b3bd...	194	-161
5	285ab9426d6982034523a855f...	194	-166
6	2fb597c2f772eca01b1f5c561b...	194	-155
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	437222e3fd1b07396f1d9ba8c...	187	-144
11	c27815f7e3dd0b926b5855262...	187	-162
12	dfe5f68118c2576143240b8d7...	186	-153

Insights: Delayed Delivery Scenario: An order took 209 days for delivery, surpassing the expected delivery time by 181 days. Possible reasons include address discrepancies, eventually resolved for successful delivery to the correct address.

Recommendations:

Enhanced Delivery Network: Increasing the number of delivery partners in a city or state has the potential to decrease delivery time, improving customer satisfaction.

Accurate Estimated Delivery: Ensuring precise estimation of delivery dates is crucial to manage customer expectations and prevent perceptions of extended delivery durations.

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

Sol.

```
 WITH freight AS
(SELECT c.customer_id,oi.freight_value,c.customer_state
FROM `sql_project.customers` AS c
LEFT JOIN `sql_project.orders` AS o
USING (customer_id)
JOIN `sql_project.order_items` AS oi
ON oi.order_id = o.order_id)
```

```
SELECT customer_state, ROUND(AVG(freight_value),2) AS average FROM freight
GROUP BY customer_state
ORDER BY average DESC
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	average
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

```
WITH freight AS
(SELECT c.customer_id,oi.freight_value,c.customer_state
FROM `sql_project.customers` AS c
LEFT JOIN `sql_project.orders` AS o
USING (customer_id)
JOIN `sql_project.order_items` AS oi
ON oi.order_id = o.order_id)

SELECT customer_state, ROUND(AVG(freight_value),2) AS average FROM freight
GROUP BY customer_state
ORDER BY average ASC
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	average
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

Insights: Freight Disparity: Notably, the state "RR" exhibits the highest average freight value, while "SP" showcases the lowest average freight among all states.

Recommendations:

State-Specific Pricing: Customize product costs based on the respective states, considering factors like taxes and regional dynamics for optimized pricing strategies.

Batched Transportation Efficiency: Implement batched transportation strategies for each state to reduce overall transportation costs and enhance operational efficiency.

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

Sol.

```
SELECT customer_state,
AVG(DATE_DIFF(
order_estimated_delivery_date,order_delivered_customer_date,hour)/24)
AS avg_delivery_time
 FROM `sql_project.orders`
JOIN `sql_project.customers`
USING(customer_id)
GROUP BY customer_state
ORDER BY avg_delivery_time DESC
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	avg_delivery_time
1	AC	20.05989583333...
2	RO	19.37894375857...
3	AP	19.04166666666...
4	AM	18.83333333333...
5	RR	16.57825203252...

```
SELECT customer_state,
AVG(DATE_DIFF(
order_estimated_delivery_date,order_delivered_customer_date,hour)/24)
AS avg_delivery_time
FROM `sql_project.orders`
JOIN `sql_project.customers`
USING(customer_id)
GROUP BY customer_state
ORDER BY avg_delivery_time ASC
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	avg_delivery_time
1	AL	8.021725440806...
2	MA	8.875697350069...
3	SE	9.313681592039...
4	ES	9.781808688387...
5	BA	10.08449785012...

Insights: Delivery Discrepancy: Customers in state AL experience quicker order deliveries compared to customers in state AC, who face longer wait times. Identifying and addressing the root causes can enhance overall delivery efficiency.

Recommendations:

Expediting Delivery: Augmenting the number of delivery partners in a state holds the potential to reduce delivery times, enhancing overall customer satisfaction.

d. 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.

Sol.

```
SELECT customer_state,
ROUND(AVG(DATE_DIFF(
order_estimated_delivery_date,order_delivered_customer_date,day)))
AS faster_delivery
FROM `sql_project.orders`
JOIN `sql_project.customers`
USING(customer_id)
GROUP BY customer_state
ORDER BY faster_delivery ASC
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	faster_delivery
1	AL	8.0
2	MA	9.0
3	SE	9.0
4	SP	10.0
5	BA	10.0

Insights: Exemplary Delivery Performance: Customers in AL state receive their products ahead of the estimated delivery date, showcasing efficient and prompt delivery services.

Recommendations:

Accelerated Delivery Impact: Swift delivery not only meets customer expectations but also fosters increased shopping frequency, creating a positive correlation between fast delivery and customer engagement.

6. Analysis based on the payments

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

Sol.

```
SELECT payment_type,
EXTRACT(month FROM order_purchase_timestamp) AS month ,
COUNT(*) AS no_of_orders
FROM `sql_project.orders` AS o
JOIN `sql_project.payments` AS p
USING (order_id)
GROUP BY payment_type,month
ORDER BY month
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	payment_type	month	no_of_orders
1	credit_card	1	6103
2	UPI	1	1715
3	voucher	1	477
4	debit_card	1	118
5	UPI	2	1723
6	credit_card	2	6609
7	voucher	2	424
8	debit_card	2	82
9	credit_card	3	7707
10	UPI	3	1942
11	debit_card	3	109
12	voucher	3	591

Insights: Payment Preference Insight: Monthly, customers predominantly use credit cards for shopping, often aligning with their salary credit timelines. The allure of additional discounts and expedited refund processing contributes to the popularity of credit card transactions, overshadowing the infrequent use of debit cards.

Recommendations:

Expansive Credit Card Discounts: Elevate credit card discounts to captivate a broader audience, enhancing the appeal of this payment mode.

Strategic Bank Partnerships: Forge advantageous partnerships with banks offering credit cards, securing mutually beneficial deals to include their cards during sales.

Payment Mode Versatility: Prioritize seamless transactions by accepting payments through various modes, ensuring flexibility and convenience for customers.

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

Sol.

```
SELECT payment_installments, COUNT(*) AS no_of_orders
FROM `sql_project.payments` AS p
GROUP BY payment_installments
HAVING payment_installments > 0
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW

Row	payment_installment	no_of_orders
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
11	11	23
12	12	133

Insights: Installment Completion Trend: A majority of products follow a pattern where customers progressively complete each installment over time, indicating a positive and consistent repayment behavior.

Recommendations:

Flexible Installment Options: Introduce flexible installment plans for customers, with interest-free options limited to fewer installments, ensuring enhanced payment convenience.

Informed Installment Approval: Assessing customer purchase history before approving installment plans can be a strategic move, providing tailored and beneficial options while minimizing potential risks for the company.