

Business Case: Target SQL

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018

1. 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.

```
Select column_name , DATA_TYPE
From target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
```

Query results

JOB INFORMATION		RESULTS	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	

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

```
select min(order_estimated_delivery_date) as first_date ,
       max(order_estimated_delivery_date) as last_date
from `target.orders`
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	first_date	last_date		
1	2016-09-30 00:00:00 UTC	2018-11-12 00:00:00 UTC		

Analysis – From the brief of this project, we are aware that the dataset focuses on the operations of the retail outlet orders placed between 2016 and 2018. With the help of the above code, we are able to identify the date and time of first and last order placed for this outlet

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

```
select count(distinct(geolocation_city)) as city,  
       count(distinct(geolocation_state)) as state  
from `target.geolocation`
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	city	state	
1	8011	27	

Analysis: Our customer presence is in 27 States and 8011 Cities.

In-depth Exploration:

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

```
select extract(Year from order_purchase_timestamp) as order_year, count(*) as  
total_orders  
from `target.orders`  
group by 1  
order by 1
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	order_year	total_orders	
1	2016	329	
2	2017	45101	
3	2018	54011	

Analysis: Basis above we can identify the total no. of orders placed in each Year. We can see that there is a growth in total number of orders placed Year on Year. In 2016, we have 329 orders, and in 2017 the count of orders has increased drastically, total 45101. In 2018 as well, the count of orders has increased from previous year, total 54011.

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

```
SELECT count(*) as orders_per_monthandyear,  
EXTRACT(YEAR FROM order_purchase_timestamp) as purchase_year,  
EXTRACT(MONTH FROM order_purchase_timestamp) as purchase_month  
from `target.orders`  
group by purchase_month,purchase_year  
order by purchase_year ,purchase_month
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	orders_per_monthandyear	purchase_year	purchase_month	
1	4	2016	9	
2	324	2016	10	
3	1	2016	12	
4	800	2017	1	
5	1780	2017	2	
6	2682	2017	3	
7	2404	2017	4	
8	3700	2017	5	
9	3245	2017	6	
10	4026	2017	7	

Analysis: in 2016 oct has highest number of orders ,In 2017 july has highest number of orders .

2. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
 1. 0-6 hrs : Dawn
 2. 7-12 hrs : Mornings
 3. 13-18 hrs : Afternoon
 4. 19-23 hrs : 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 time_of_day,  
count(*) as orders_per_monthandyear,
```

```

from `target.orders`
group by time_of_day
order by time_of_day

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	time_of_day ▼	orders_per_month		
1	Afternoon	38135		
2	Dawn	5242		
3	Morning	27733		
4	Night	28331		

Analysis: Basis the query results, we can understand that the majority of the orders placed by customers in Brazil are in Afternoon i.e. between 13:00 hours and 18:00 hours

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

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

```

with CTE as
(select c.customer_state, o.order_id, extract(Year from
o.order_purchase_timestamp) as Year, extract(Month from o.order_purchase_timestamp)
as Month
from `Target.orders` o
left join `Target.customers` c
on c.customer_id = o.customer_id)
select customer_state, Year, Month, count(order_id) as Order_count, count(order_id) -
(lag(count(order_id)) over(partition by customer_state order by year, Month)) as
Month_on_Month
from CTE
group by 1, 2, 3

```

Query results						
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH
Row	customer_state	Year	Month	Order_count	Month_on_Month	
1	MS	2017	1	1	null	
2	MS	2017	2	11	10	
3	MS	2017	3	20	9	
4	MS	2017	4	15	-5	
5	MS	2017	5	29	14	
6	MS	2017	6	27	-2	
7	MS	2017	7	25	-2	
8	MS	2017	8	24	-1	
9	MS	2017	9	33	9	
10	MS	2017	10	34	1	

Analysis: We can see that there is an increase and then substantial decrease in the number of orders month on month for each state.

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

```
select customer_state, count(customer_id) as customer_number
from `target.customers`
group by 1
order by 1
```

Query results			
JOB INFORMATION		RESULTS	EXECUTION DETAILS
Row	customer_state	customer_number	
1	AC	81	
2	AL	413	
3	AM	148	
4	AP	68	
5	BA	3380	
6	CE	1336	
7	DF	2140	
8	ES	2033	
9	GO	2020	
10	MA	747	

Analysis: state BA has the highest number of customers.

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

1. 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.

```
with cte1 as( select * from `target.orders` as o
join `target.payments` as p
on o.order_id=p.order_id
where extract(year from order_purchase_timestamp) between 2017 and 2018 and extract
(month from order_purchase_timestamp) between 1 and 8 ),
cte2 as( select extract(year from order_purchase_timestamp) as year,
sum(payment_value) as cost from cte1 group by year )
select *, round((((cost)-lag(cost)over(order by year))*100)/lag(cost)over(order by
year),2)as percentage_increase from cte2
order by year desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	year ▼	cost ▼	percentage_increase	
1	2018	8694733.839999...	136.98	
2	2017	3669022.119999...	null	

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

```
select c.customer_state, round(sum(ot.price),2) as total_price ,
round(avg(ot.price),2) as avg_price
from `target.customers` c
join
`target.orders` o
on o.customer_id = c.customer_id
join
`target.order_items` ot
on ot.order_id = o.order_id
group by customer_state
order by 1
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	
Row	customer_state		total_price	avg_price	
1	AC		15982.95	173.73	
2	AL		80314.81	180.89	
3	AM		22356.84	135.5	
4	AP		13474.3	164.32	
5	BA		511349.99	134.6	
6	CE		227254.71	153.76	
7	DF		302603.94	125.77	
8	ES		275037.31	121.91	
9	GO		294591.95	126.27	
10	MA		119648.22	145.2	

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

```
select c.customer_state, round(sum(ot.freight_value),2) as total_freight ,
       round(avg(ot.freight_value),2) as avg_freight
from `target.customers` c
join
`target.orders` o
on o.customer_id = c.customer_id
join
`target.order_items` ot
on ot.order_id = o.order_id
group by customer_state
order by 1
```

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state ▼	total_freight ▼	avg_freight ▼	
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	
7	DF	50625.5	21.04	
8	ES	49764.6	22.06	
9	GO	53114.98	22.77	
10	MA	31523.77	38.26	

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

- 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_estimated_delivery_date - order_delivered_customer_date

```
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 `target.orders`
```


Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	order_id	time_to_deliver	diff_estimated_delive	
1	1950d777989f6a877539f5379...	30	-12	
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28	
3	65d1e226dfaeb8cdc42f66542...	35	16	
4	635c894d068ac37e6e03dc54e...	30	1	
5	3b97562c3aee8bdecb5c2e45...	32	0	
6	68f47f50f04c4cb6774570cfde...	29	1	
7	276e9ec344d3bf029ff83a161c...	43	-4	
8	54e1a3c2b97fb0809da548a59...	40	-4	
9	fd04fa4105ee8045f6a0139ca5...	37	-1	
10	302bb8109d097a9fc6e9cefc5...	33	-5	

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

```
(select s.seller_state, round(avg(oi.freight_value),2) as avg_freight
from `target.order_items` oi
join `target.sellers` s
on s.seller_id = oi.seller_id
group by 1
order by 2
limit 5)
```

union all

```
(select s.seller_state, round(avg(oi.freight_value),2) as avg_freight
from `target.order_items` oi
join `target.sellers` s
on s.seller_id = oi.seller_id
group by 1
order by 2 desc
limit 5)
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	seller_state ▼	avg_freight ▼		
1	SP	18.45		
2	PA	19.39		
3	RJ	19.47		
4	DF	20.57		
5	PR	22.72		
6	RO	50.91		
7	CE	46.38		
8	PB	39.19		
9	PI	36.94		
10	AC	32.84		

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

```
(select c.customer_state,
       round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,Day)),0)
as time_to_deliver
  from `target.orders` o
 inner join `target.customers` c
  on o.customer_id = c.customer_id
 group by 1
  order by 2
  limit 5)
union all
```

```
(select c.customer_state,
       round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,Day)),0)
as time_to_deliver
  from `target.orders` o
 inner join `target.customers` c
  on o.customer_id = c.customer_id
 group by 1
  order by 2 desc
  limit 5)
```

Query results			
JOB INFORMATION		RESULTS	JSON
Row	customer_state	time_to_deliver	EXECUTION DETAILS
1	SP	8.0	
2	MG	12.0	
3	PR	12.0	
4	DF	13.0	
5	SC	14.0	
6	RR	29.0	
7	AP	27.0	
8	AM	26.0	
9	AL	24.0	
10	PA	23.0	

- 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.

```

select c.customer_state ,
round(avg(date_diff(o.order_estimated_delivery_date , o.order_delivered_customer_date ,
day)),0) as delivery_date
from `target.orders` o
join
`target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2
limit 5

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state ▼	delivery_date ▼		
1	AL	8.0		
2	MA	9.0		
3	SE	9.0		
4	SP	10.0		
5	BA	10.0		

6. Analysis based on the payments:

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

```
select count(o.order_id) as number_of_order ,
       EXTRACT(MONTH FROM o.order_purchase_timestamp) as month,
       EXTRACT(Year FROM o.order_purchase_timestamp) as Year,
       p.payment_type
from `target.orders` o
join
`target.payments` p
on o.order_id = p.order_id
group by 2,3,4
order by 1 desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	number_of_order ▼	month ▼	Year ▼	payment_type ▼	
1	5897	11	2017	credit_card	
2	5691	3	2018	credit_card	
3	5520	1	2018	credit_card	
4	5497	5	2018	credit_card	
5	5455	4	2018	credit_card	
6	5253	2	2018	credit_card	
7	4985	8	2018	credit_card	
8	4813	6	2018	credit_card	
9	4755	7	2018	credit_card	
10	4377	12	2017	credit_card	

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

```
select count(order_id) as no_of_orders ,  
       payment_installments  
from `target.payments`  
group by 2  
order by 1 desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	no_of_orders	payment_installment	
1	52546	1	
2	12413	2	
3	10461	3	
4	7098	4	
5	5328	10	
6	5239	5	
7	4268	8	
8	3920	6	
9	1626	7	
10	644	9	

7. Actionable Insights & Recommendations (10 points)

3942 UNTAPPED CITIES:

- **Market Research and Analysis:**

Conduct thorough market research and analysis of the untapped cities to understand the potential customer base, market demand, competition, and infrastructure.

- **Prioritization Strategy:** Prioritize the untapped cities based on factors like population density, economic growth, market size, and alignment with your target customer segment. Focus on cities that offer a balance between market potential and operational feasibility.

- **Localized Customer Support:** Invest in localized customer support capabilities for the untapped cities. Hire customer support representatives who are familiar with the culture and nuances of each city to provide personalized and efficient assistance.

- **Flexible Pricing and Payment Options:** Consider adjusting the pricing strategies and offering flexible payment options to cater to the market dynamics and affordability in each untapped city. Offer a range of payment options that are commonly used and preferred by customers in each city. MAJORITY CUSTOMER PRESENCE IN 'SP', 'RJ', 'MG':

- **Localized Product Offerings:** Analyze the preferences and buying patterns of customers in 'SP', 'RJ', and 'MG' to tailor your product offerings.

- **Localized Product Offerings:** Analyze the preferences and buying patterns of customers in 'SP', 'RJ', and 'MG' to tailor your product offerings.

- **Partnerships and Collaborations:** Identify potential partnerships or collaborations with local businesses, influencers, or organizations in 'SP', 'RJ', and 'MG'. Leverage their networks and credibility to expand your customer base and strengthen your presence in these states. Co-marketing campaigns or joint promotional activities can help increase brand awareness and establish a stronger foothold. TARGET AUDIENCE:

- **Optimize Inventory:** Since the afternoon is the peak period for order placements, ensure that your inventory levels are well-stocked during this time.

- **Staffing:** Analyze historical data to determine the most popular products and adjust inventory accordingly to avoid stockouts. Additionally, ensure you have sufficient staff available to handle the increased order volume and provide timely customer support.

- **Personalized Marketing Campaigns:** Leverage the knowledge of peak order times to create targeted marketing campaigns. Send timely and relevant promotional offers, discounts, or personalized recommendations to customers during the afternoon hours. Use customer segmentation and purchase history data to tailor your marketing messages and increase the chances of converting customers during this peak period