

## Business Case: Target SQL

Description: 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. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

We are trying to analyze this extensive dataset, so that we can gain valuable insights into Target's operations in Brazil

I. 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.

Query

```
SELECT column_name, data_type FROM `Target`.INFORMATION_SCHEMA.COLUMNS
where table_name="customers"
```

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUT
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		

Insight: Only customer\_zip\_code is in integer datatype and other columns are of string datatype.

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

Query

```
SELECT date_diff(max(order_purchase_timestamp), min(order_purchase_timestamp), day) as
Time_difference FROM `Target.orders`
```

## Query results

JOB INFORMATION		RESULTS
Row	Time_difference	
1	772	

Insight: orders were placed within 772 days.

### 3.Count the number of Cities and States in our dataset.

#### Query

```
SELECT count(distinct geolocation_city) as city_count, count(distinct geolocation_state) as count_state FROM `Target.geolocation`
```

## Query results

JOB INFORMATION		RESULTS	JSON
Row	city_count	count_state	
1	8011	27	

Insight:total number of cities and states were present in geolocation Table and there were total 8011 cities and 27 states.

## II.In-depth Exploration

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

#### Query

```
select * from (SELECT count(order_id)as order_sum,extract(month from order_purchase_timestamp)as month,extract(year from order_purchase_timestamp) as year FROM `Target.orders` group by extract(month from order_purchase_timestamp), extract(year from order_purchase_timestamp) )as x order by x.year,x.month
```

## Query results

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

Insight: Number of order placed have increased over the years  
Minimum number of orders were placed in December 2016  
Maximum number of orders were placed in November 2017

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

### Query

```
select * from (SELECT count(order_id)as order_sum,extract(month from
order_purchase_timestamp)as month, FROM `Target.orders` group by extract(month from
order_purchase_timestamp) )as x
order by x.month
```

## Query results

JOB INFORMATION		RESULTS	JSON
Row	order_sum	month	
1	8069	1	
2	8508	2	
3	9893	3	
4	9343	4	
5	10573	5	
6	9412	6	
7	10318	7	
8	10843	8	
9	4305	9	
10	4959	10	

Insight: Lowest number of orders were placed in the month of September.

Highest number of orders were placed in the month of August.

3. 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 x.Timings, count(order_id) as total_order from
  (Select order_id,
   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 "Mornings"
   when extract(hour from order_purchase_timestamp) between 13 and 18 then "Afternoon"
   else "Night"
  end as Timings FROM `Target.orders`) x
group by x.Timings
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTI
Row	Timings		total_order	
1	Mornings		27733	
2	Dawn		5242	
3	Afternoon		38135	
4	Night		28331	

Insight: Brazilian customers place most of their orders in the Afternoon and least Amount of orders are placed at Dawn.

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

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

Query:

```
Select * from(SELECT count(o.order_id) as count_order,c.customer_state, extract(month
from order_purchase_timestamp) as month FROM `Target.orders` as o left join
`Target.customers` as c on o.customer_id=c.customer_id group by
c.customer_state,extract(month from order_purchase_timestamp))x order by
x.customer_state,x.month
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	count_order	customer_state	month	
1	8	AC	1	
2	6	AC	2	
3	4	AC	3	
4	9	AC	4	
5	10	AC	5	
6	7	AC	6	
7	9	AC	7	
8	7	AC	8	
9	5	AC	9	
10	6	AC	10	

Insight: Maximum number of orders were placed from state SP.

Least number of orders were placed from state RR

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

Query

```
SELECT count(distinct c.customer_id) as count_Customer,c.customer_state, FROM
`Target.orders` as o left join `Target.customers` as c on o.customer_id=c.customer_id
group by c.customer_state
```

## Query results

JOB INFORMATION		RESULTS	JSON
Row	count_Customer	customer_state	
1	12852	RJ	
2	5466	RS	
3	41746	SP	
4	2140	DF	
5	5045	PR	
6	907	MT	
7	747	MA	
8	413	AL	
9	11635	MG	
10	1652	PE	

Insight: State RR has the least number of customers

State SP is home to maximum number of customers.

IV.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.

Query

```
Select *, (((y.cost-y.prev_cost)/y.prev_cost)*100) as percentage_inc
from (Select x.year, sum(x.payment_value) as cost,
lag(sum(x.payment_value),1) over(order by x.year) as prev_cost
from (SELECT
    extract(year from o.order_purchase_timestamp) as year,
    extract(month from o.order_purchase_timestamp) as month,
    p.payment_value
FROM `Target.orders` as o left join `Target.payments` as p
on o.order_id=p.order_id )x
where (x.year between 2017 and 2018) and (x.month between 1 and 8)
group by x.year) y
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXEC
Row	year ▼	cost ▼	prev_cost ▼	percentage_inc ▼	
1	2017	3669022.119999...	null	null	
2	2018	8694733.839999...	3669022.119999...	136.9768716466...	

Insight: There is an increase of 137 percent from 2017 to 2018 in cost of orders.

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

Query

```
SELECT c.customer_state,
       sum(p.payment_value) as total,
```

```

avg(p.payment_value) as avg
FROM `Target.customers` as c left join `Target.orders` as o on c.customer_id
=o.customer_id left join `Target.payments`
as p on o.order_id=p.order_id
group by c.customer_state

```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	total	avg	
1	RN	102718.13	196.7780268199...	
2	CE	279464.0299999...	199.90273962804	
3	RS	890898.5399999...	157.1804057868...	
4	SC	623086.4299999...	165.9793367075...	
5	SP	5998226.959999...	137.5046297739...	
6	MG	1872257.260000...	154.7064336473...	
7	BA	616645.8200000...	170.8160166204...	
8	RJ	2144379.689999...	158.5258882235...	
9	GO	350092.3099999...	165.7634043560...	
10	MA	152523.0200000...	198.8566101694...	

Insight: State SP has the highest Total value and state PB has the highest average value.

State RR has the lowest total value and state SP has the lowest average value.

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

Query

```

SELECT c.customer_state,
sum(oi.freight_value) as total_freight,
avg(oi.freight_value) as avg_freight
FROM `Target.customers` as c left join `Target.orders` as o on c.customer_id
=o.customer_id left join
`Target.order_items` as oi
on o.order_id=oi.order_id
group by c.customer_state

```



## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state ▼	total_freight ▼	avg_freight ▼	
1	RN	18860.09999999...	35.65236294896...	
2	CE	48351.58999999...	32.71420162381...	
3	RS	135522.7400000...	21.73580433039...	
4	SC	89660.26000000...	21.47036877394...	
5	SP	718723.0699999...	15.14727539041...	
6	MG	270853.4600000...	20.63016680630...	
7	BA	100156.6799999...	26.36395893656...	
8	RJ	305589.3100000...	20.96092393168...	
9	GO	53114.97999999...	22.76681525932...	
10	MA	31523.77000000...	38.25700242718...	

**Insight:** State SP has the highest total freight value and state RR has the highest average freight value.

State RR has the lowest total freight value and state SP has the lowest average freight value.

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

1. 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:

- $$\text{time\_to\_deliver} = \text{order\_delivered\_customer\_date} - \text{order\_purchase\_timestamp}$$

b.  $\text{diff\_estimated\_delivery} = \text{order\_estimated\_delivery\_date} - \text{order\_delivered\_customer\_date}$

Query

```
Select order_id,
date_diff( order_delivered_customer_date,order_purchase_timestamp, day)as
days_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_days
from `Target.orders`
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	order_id	days_to_deliver	diff_days	
1	1950d777989f6a877539f5379...	30	-12	
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28	
3	65d1e226dfaeb8cdc42f66542...	35	16	
4	635c894d068ac37e6e03dc54e...	30	1	
5	3b97562c3aee8bdedcb5c2e45...	32	0	
6	68f47f50f04c4cb6774570cfde...	29	1	
7	276e9ec344d3bf029ff83a161c...	43	-4	
8	54e1a3c2b97fb0809da548a59...	40	-4	
9	fd04fa4105ee8045f6a0139ca5...	37	-1	
10	302bb8109d097a9fc6e9cefc5...	33	-5	

Insight: Target takes an average of 13 days to deliver and average difference of days in estimated and actual delivery time is 11 days.

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

Query

```
select *, dense_rank()over(order by avg_freight asc) as top_5, 0 as bottom_five from
(select x. customer_state,x.avg_freight
from (SELECT c.customer_state,
avg(oi.freight_value) as avg_freight
```

```

FROM `Target.customers` as c left join `Target.orders` as o on c.customer_id
=o.customer_id left join
  `Target.order_items` as oi
  on o.order_id=oi.order_id
group by c.customer_state)x order by avg_freight limit 5 offset 22 )

union all
select *,0 as top_five, dense_rank()over(order by avg_freight ) as bottom_five
from(select x. customer_state,x.avg_freight
from (SELECT c.customer_state,
  avg(oi.freight_value) as avg_freight
FROM `Target.customers` as c left join `Target.orders` as o on c.customer_id
=o.customer_id left join
  `Target.order_items` as oi
  on o.order_id=oi.order_id
group by c.customer_state)x order by avg_freight limit 5)

```

## Query results



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	
Row	customer_state ▼	avg_freight ▼	top_5 ▼	bottom_five ▼			
1	PR	20.53165156794...	0	2			
2	RJ	20.96092393168...	0	4			
3	MG	20.63016680630...	0	3			
4	DF	21.04135494596...	0	5			
5	SP	15.14727539041...	0	1			
6	PI	39.14797047970...	1	0			
7	RO	41.06971223021...	3	0			
8	RR	42.98442307692...	5	0			
9	AC	40.07336956521...	2	0			
10	PB	42.72380398671...	4	0			

Insight: Top state with highest freight value is RR and top state with lowest freight value is DF.

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

Query

```
select *, dense_rank()over(order by avg_days asc) as top_5, 0 as bottom_five from
(select x.customer_state, avg(days_to_deliver) avg_days from(SELECT c.customer_state,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp, day)as
days_to_deliver FROM `Target.orders` as o left join `Target.customers` as c on
o.customer_id=c.customer_id ) x group by x.customer_state order by avg_days limit 5
offset 22)
```

union all

```
select *,0 as top_five, dense_rank()over(order by avg_days ) as bottom_five
from
(select x.customer_state, avg(days_to_deliver) avg_days from(SELECT c.customer_state,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp, day)as
days_to_deliver FROM `Target.orders` as o left join `Target.customers` as c on
o.customer_id=c.customer_id ) x group by x.customer_state order by avg_days limit 5 )
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAI
Row	customer_state	avg_days	top_5	bottom_five	
1	SP	8.298061489072...	0	1	
2	DF	12.50913461538...	0	4	
3	PR	11.52671135486...	0	2	
4	SC	14.47956019171...	0	5	
5	MG	11.54381329810...	0	3	
6	AL	24.04030226700...	2	0	
7	PA	23.31606765327...	1	0	
8	RR	28.97560975609...	5	0	
9	AM	25.98620689655...	3	0	
10	AP	26.73134328358...	4	0	

Insight: Top state with highest average delivery days is RR and top state with lowest average delivery days is SC.

4. 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 *, dense_rank()over(order by y.delivery_speed) from (select x.
customer_state, (avg(actual_days)-avg(estimated_days))as delivery_speed from(SELECT
c.customer_state,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp, day)as
actual_days,
date_diff(o.order_estimated_delivery_date,o.order_purchase_timestamp, day)as
estimated_days
FROM `Target.orders` as o left join `Target.customers` as c on
o.customer_id=c.customer_id )x group by x.customer_state )y order by delivery_speed
limit 5
```

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	
Row	customer_state	delivery_speed	f0_		
1	AC	-20.1279320987...		1	
2	RO	-19.4935343775...		2	
3	AP	-18.9745390693...		3	
4	AM	-18.7705498602...		4	
5	RR	-17.1983032873...		5	

Insight: State AC has the fastest delivery.

VI. Analysis based on the payments:

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

Query

```
select * from(SELECT count(p.order_id) as count_order,p.payment_type, extract(month
from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year FROM `Target.payments` as p left
join `Target.orders` as o
```

```
on p.order_id=o.order_id group by p.payment_type, extract(year from
o.order_purchase_timestamp),extract(month from o.order_purchase_timestamp))x
order by x.payment_type,x.year, x.month
```

## Query results



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH
Row	count_order	payment_type		month	year	
1	63	UPI		10	2016	
2	197	UPI		1	2017	
3	398	UPI		2	2017	
4	590	UPI		3	2017	
5	496	UPI		4	2017	
6	772	UPI		5	2017	
7	707	UPI		6	2017	
8	845	UPI		7	2017	
9	938	UPI		8	2017	
10	903	UPI		9	2017	

Insight: According to the data the least number of orders were paid for in the month of december 2016 using credit card.

The most number of orders were paid for using credit card in the month of november 2017

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

Query

```
select count(order_id) as count_order, x.payment_installments from(SELECT *,
FROM `Target.payments`
where payment_sequential >=1)x group by x.payment_installments
```

## Query results

JOB INFORMATION		RESULTS	JSON
Row	count_order	payment_installment	
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	

Insight: Least number of orders were placed for emi installments of >20.

Maximum number of installments were placed for emi installments of <5.

## Recommendations

1. In brazil target should introduce some offers or discounts during the month of september and october to boost the sale during those months.
2. Target can introduce special discounts or sales or fresh consumables during dawn to increase number of order being placed during dawn.
3. Should introduce first time purchase offers so that many people from other states also shop from target.
4. Target should adopt methods to decrease the average difference between estimated delivery date and actual delivery date to lessen unsatisfied customers due to late delivery.

