Target: Project Using SQL(BigQuery)

Context

Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb

Q 1: Usual exploratory analysis steps like checking the structure & characteristics of the dataset

I. Data type of columns in a table

```
SELECT  column_name, data_type
FROM `target-brazil.Target_My_SQL_Project`.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers'
```

Query results

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

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

```
SELECT  column_name, data_type
```

```
FROM `target-brazil.Target_My_SQL_Project`.INFORMATION_SCHEMA.COLUMNS
where table_name = 'geolocation'
```

Query results

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

Row	column_name	data_type
1	geolocation_zip_code_prefix	INT64
2	geolocation_lat	FLOAT64
3	geolocation_lng	FLOAT64
4	geolocation_city	STRING
5	geolocation_state	STRING

```
SELECT  column_name, data_type
FROM `target-brazil.Target_My_SQL_Project`.INFORMATION_SCHEMA.COLUMNS
where table_name = 'order_items'
```

Query results

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

Row	column_name	data_type
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64

Etc……

II. Time period for which the data is given

```
SELECT MIN(order_purchase_timestamp) AS frist_order,
MAX(order_purchase_timestamp) AS last_order
FROM target-brazil.Target_My_SQL_Project.orders;
```

Query results

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

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

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

```
SELECT COUNT(DISTINCT geolocation_city) AS cities,
COUNT(DISTINCT geolocation_state) AS states
FROM target-brazil.Target_My_SQL_Project.geolocation;
```

Query results

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

Row	cities	states
1	8011	27

Q 2: In-depth Exploration:

I. 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?

```
SELECT EXTRACT(year FROM order_purchase_timestamp) AS year,
  EXTRACT(month from order_purchase_timestamp) AS month,
  count(*) AS Total_order
FROM `target-brazil.Target_My_SQL_Project`.orders
GROUP BY year, month
ORDER BY total_order desc;
```

Query results

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

Row	year	month	Total_order
1	2017	11	7544
2	2018	1	7269
3	2018	3	7211
4	2018	4	6939
5	2018	5	6873
6	2018	2	6728
7	2018	8	6512
8	2018	7	6292
9	2018	6	6167
10	2017	12	5673
11	2017	10	4631

Remark: **There is no specific month in which the order value is very high compared to the other months. Hence, we do not see a trend here.**

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

```
SELECT sum(total_order) AS Order_value,
CASE WHEN hour BETWEEN 5 and 8 THEN 'dawn'
  WHEN hour BETWEEN 8 and 12 THEN 'morning'
  WHEN hour BETWEEN 13 and 17 THEN 'afternoon'
  ELSE 'night'
  END AS time_of_buy
FROM (
SELECT EXTRACT(hour FROM order_purchase_timestamp) AS hour,
  COUNT(*) AS Total_order
FROM `target-brazil.Target_My_SQL_Project`.orders
GROUP BY hour
ORDER BY hour) x
GROUP BY time_of_buy
ORDER BY Order_value;
```

Query results

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

Row	Order_value	time_of_buy
1	4888	dawn
2	23535	morning
3	32366	afternoon
4	38652	night

Remark: We clearly see that there is a trend. Order value is the highest in night, followed by afternoon and then morning.

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

I. Getting month on month orders by states

```
SELECT EXTRACT(month FROM order_purchase_timestamp) AS Month,
EXTRACT(year FROM order_purchase_timestamp) AS Year,
customer_state,
COUNT(*) AS num_order
FROM `target-brazil.Target_My_SQL_Project`.orders o
INNER JOIN `target-brazil.Target_My_SQL_Project`.customers c
ON o.customer_id = c.customer_id
GROUP BY customer_state, Year, Month
ORDER BY num_order DESC;
```

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

Row	Month	Year	customer_state	num_order
1	8	2018	SP	3253
2	5	2018	SP	3207
3	4	2018	SP	3059
4	1	2018	SP	3052
5	3	2018	SP	3037
6	11	2017	SP	3012
7	7	2018	SP	2777
8	6	2018	SP	2773
9	2	2018	SP	2703
10	12	2017	SP	2357

Remark: **State ‘SP’ has the highest number of orders among all the states in almost every single month.**

II. Distribution of customers across the states in Brazil

```
SELECT customer_state,
COUNT(customer_id) AS cx_every_state
FROM `target-brazil.Target_My_SQL_Project`.customers
GROUP BY customer_state
ORDER BY cx_every_state DESC;
```

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

Row	customer_state	cx_every_state
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

Remark: **State ‘SP’ has the highest number of customers followed by ‘RJ’ and ‘MG’.**

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

I. 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

```
WITH cte1 AS (
  SELECT
  EXTRACT(year FROM order_purchase_timestamp) AS year,
  SUM(payment_value) AS cost
```

```
  FROM `target-brazil.Target_My_SQL_Project`.orders o
  JOIN `target-brazil.Target_My_SQL_Project`.payments p
  ON o.order_id = p.order_id
  WHERE EXTRACT(month FROM order_purchase_timestamp) BETWEEN
1 AND 8
  AND EXTRACT(year FROM order_purchase_timestamp) BETWEEN 2017
AND 2018
  GROUP BY year
  ORDER BY year ),
cte2 AS (
  SELECT *,
  LAG(cost, 1) OVER(ORDER BY year) AS last_year_cost
  FROM cte1 )
SELECT *, (cost -
 COALESCE(last_year_cost, 0))/ COALESCE(cost, 1)*100 AS difference
FROM cte2
ORDER BY year;
```

Query results

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

Row	year	cost	last_year_cost	difference
1	2017	3669022.11...	*null*	100.0
2	2018	8694733.83...	3669022.11...	57.8017891...

Remark: **From 2017 to 2018, there has been an increase of 57% in order value.**

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

```
SELECT customer_state, AVG(price) AS Mean_price, AVG(freight_value) AS Mean_frei
ght,
SUM(price) AS Total_price, SUM(freight_value) AS Total_freight
FROM `target-brazil.Target_My_SQL_Project`.order_items oi
JOIN `target-brazil.Target_My_SQL_Project`.orders o
ON o.order_id = oi.order_id
JOIN `target-brazil.Target_My_SQL_Project`.customers c
ON o.customer_id=c.customer_id
GROUP BY customer_state
```

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

Row	customer_state	Mean_price	Mean_freight	Total_price	Total_freight
1	SP	109.653629...	15.1472753...	5202955.05...	718723.069...
2	RJ	125.117818...	20.9609239...	1824092.66...	305589.310...
3	PR	119.004139...	20.5316515...	683083.760...	117851.680...
4	SC	124.653577...	21.4703687...	520553.340...	89660.2600...
5	DF	125.770548...	21.0413549...	302603.939...	50625.4999...
6	MG	120.748574...	20.6301668...	1585308.02...	270853.460...
7	PA	165.692416...	35.8326851...	178947.809...	38699.3000...
8	BA	134.601208...	26.3639589...	511349.990...	100156.679...
9	GO	126.271731...	22.7668152...	294591.949...	53114.9799...
10	RS	120.337453...	21.7358043...	750304.020...	135522.740...
11	TO	157.529333...	37.2466031...	49621.7400...	11732.6799...

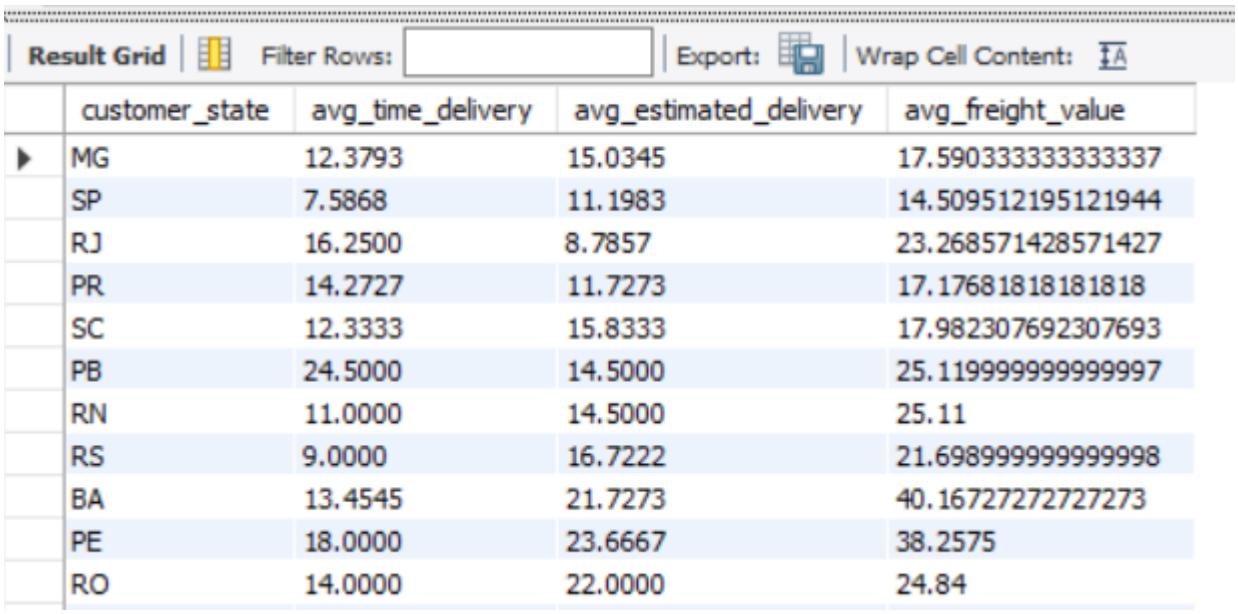
Remark: **Averge price & avg freight value; Total Price & total Freight value for all 27 states.**

Q 5: Analysis on sales, freight and delivery time

I. **Calculate days between purchasing, delivering and estimated delivery**
II. **Find time_to_delivery & diff_estimated_delivery.**

```
with base1 as
(select customer_state, freight_value,
datediff(order_delivered_customer_date, order_purchase_timestamp) AS time_to
_delivery,
datediff(order_estimated_delivery_date, order_delivered_customer_date) AS diff
_estimated_delivery
from order_items oi
join orders o
ON o.order_id = oi.order_id
JOIN customers c
on o.customer_id = c.customer_id
)

select customer_state, avg(time_to_delivery) as avg_time_delivery,
avg(diff_estimated_delivery) as avg_estimated_delivery,
avg(freight_value) as avg_freight_value
from base1
group by customer_state;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_state	avg_time_delivery	avg_estimated_delivery	avg_freight_value
MG	12.3793	15.0345	17.590333333333337
SP	7.5868	11.1983	14.509512195121944
RJ	16.2500	8.7857	23.268571428571427
PR	14.2727	11.7273	17.176818181818
SC	12.3333	15.8333	17.982307692307693
PB	24.5000	14.5000	25.119999999999997
RN	11.0000	14.5000	25.11
RS	9.0000	16.7222	21.698999999999998
BA	13.4545	21.7273	40.16727272727273
PE	18.0000	23.6667	38.2575
RO	14.0000	22.0000	24.84

III. Sort the data to get the following:

Top 5 states with highest/lowest average freight value -

```
with base1 as
(select customer_state, freight_value,
datediff(order_delivered_customer_date, order_purchase_timestamp) AS time_to
_delivery,
datediff(order_estimated_delivery_date, order_delivered_customer_date) AS diff
_estimated_delivery
from order_items oi
join orders o
ON o.order_id = oi.order_id
JOIN customers c
on o.customer_id = c.customer_id)
select customer_state, avg(time_to_delivery) as avg_time_delivery,
avg(diff_estimated_delivery) as avg_estimated_delivery,
avg(freight_value) as avg_freight_value
from base1
group by customer_state
order by avg_freight_value DESC
limit 5;
```

	customer_state	avg_time_delivery	avg_estimated_delivery	avg_freight_value
▶	BA	13.4545	21.7273	40.16727272727273
	PE	18.0000	23.6667	38.2575
	RR	30.0000	17.0000	33.24
	CE	17.0000	13.0000	32.129999999999995
	ES	12.5000	14.0000	29.55

```
with base1 as
(select customer_state, freight_value,
datediff(order_delivered_customer_date, order_purchase_timestamp) as time_to
_delivery,
datediff(order_estimated_delivery_date, order_delivered_customer_date) as diff_
estimated_delivery
from order_items oi
join orders o
on o.order_id = oi.order_id
join customers c
on o.customer_id = c.customer_id)
select customer_state, avg(time_to_delivery) as avg_time_delivery,
avg(diff_estimated_delivery) as avg_estimated_delivery,
avg(freight_value) as avg_freight_value
from base1
group by customer_state
order by avg_freight_value
limit 5;
```

	customer_state	avg_time_delivery	avg_estimated_delivery	avg_freight_value
▶	SP	7.5868	11.1983	14.509512195121944
	PR	14.2727	11.7273	17.176818181818
	DF	9.5000	11.5000	17.385
	MG	12.3793	15.0345	17.590333333333337
	SC	12.3333	15.8333	17.982307692307693

IV. Top 5 states with highest/lowest average time to delivery

Lowest:

	customer_state	avg_time_delivery	avg_estimated_delivery	avg_freight_value
▶	SP	7.5868	11.1983	14.509512195121944
	RS	9.0000	16.7222	21.698999999999998
	DF	9.5000	11.5000	17.385
	GO	10.6000	16.6000	27.248
	RN	11.0000	14.5000	25.11

Highest:

	customer_state	avg_time_delivery	avg_estimated_delivery	avg_freight_value
▶	TO	31.0000	0.0000	28.95
	RR	30.0000	17.0000	33.24
	AP	26.0000	28.0000	26.36
	SE	25.0000	-1.0000	22.85
	PB	24.5000	14.5000	25.119999999999997

V. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Fast delivery, compared to estimated date:

```
select customer_state, freight_value,
datediff(order_estimated_delivery_date, order_delivered_customer_date) as diff_
estimated_delivery
from order_items oi
join orders o
on o.order_id = oi.order_id
join customers c
on o.customer_id = c.customer_id
group by customer_state
order by diff_estimated_delivery desc
limit 5;
```

	customer_state	freight_value	diff_estimated_delivery
▶	AP	26.36	28
	RO	24.84	22
	GO	19.22	18
	PA	24.84	17
	RR	33.24	17

‘Not so Fast’ delivery, compared to estimated date:

```
with base1 as(
select customer_state, freight_value,
datediff(order_estimated_delivery_date, order_delivered_customer_date) AS diff_estim
ated_delivery
from order_items oi
join orders o
ON o.order_id = oi.order_id
join customers c
on o.customer_id = c.customer_id
)
select *, coalesce(diff_estimated_delivery, 0) as non_null_estimated_delivery
from base1
group by customer_state
order by diff_estimated_delivery
limit 5;
```

customer_state	freight_value	diff_estimated_delivery	non_null_estimated_delivery
SC	15.35	NULL	0
SE	22.85	-1	-1
TO	28.95	0	0
ES	25.14	6	6
SP	8.27	7	7

6. Payment type analysis:

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

```
SELECT
  payment_type as payment_methods,
  count(o.order_id) as num_orders,
  extract(month from o.order_purchase_timestamp) as month,
  extract(year from o.order_purchase_timestamp) as year
from `target-brazil.Target_My_SQL_Project.payments` as p
join `target-brazil.Target_My_SQL_Project.orders` as o
on p.order_id = o.order_id
group by payment_methods, year, month
order by year, month;
```

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

Row	payment_methods	num_orders	month	year
1	credit_card	3	9	2016
2	credit_card	254	10	2016
3	voucher	23	10	2016
4	debit_card	2	10	2016
5	UPI	63	10	2016
6	credit_card	1	12	2016
7	voucher	61	1	2017
8	UPI	197	1	2017
9	credit_card	583	1	2017
10	debit_card	9	1	2017

II. Count of orders based on the no. of payment installments

```
SELECT
  distinct(payment_installments) as num_installment,
  count(order_id) as num_orders
from `target-brazil.Target_My_SQL_Project.payments`
where payment_installments > 1
group by num_installment
order by num_orders DESC
```

JOB INFORMATION | RESULTS | JSO

Row	num_installment	num_orders
1	2	12413
2	3	10461
3	4	7098
4	10	5328
5	5	5239
6	8	4268
7	6	3920
8	7	1626
9	9	644
10	12	133

Actionable Insights:

a) Brazilian customers tend to buy more at night after 6 p.m. it is time to be at home after finishing the professional life
b) It is witnessed that there is huge orders in particular months such as August
c) 'SP' state has the highest number of customers in Brazil followed by 'RJ', 'MG' and 'RS' with 1746, 12852, 1635 and 5466 customers respectively
d) From 2017 to 2018, the number of orders has increased highly in almost entire region of Brazil
e) There are a few states which has been generating higher than others Total Order Value and Total Freight Value even when they do not have the higher number of customers
f) Target's logistics department has been able to maintain fast/before estimated delivery in a few states whereas failed to do so in few other states comparatively
g) Payment method Credit Card is opted by the customers of Brazil for most of the orders and other prominent payment method is UPI
h) Less number of payment instalments has more number of orders associated

Recommendations:

a) The company should focus on showcasing more offers and discounts plus combo products suggestion in the night time when traffic on website is the highest
b) First recommendation should be applied to particular months as well based on the number of orders and order value
c) 'SP' has the highest number of customers which means easy access for delivery and it gives an opportunity to be cost effective. Hence reducing the price and giving more offers
d) Start looking for all the factors which re thankfully the reason of such huge increase in orders. It should be continue with more lucrative additions
e) Those states with higher order value have the potential to give more customers and revenue. They are to be targeted with extra focus
f) The company should provide more offers on credit card payment since most of the customers are using Credit Card. Same goes for payment instalments. There needs to be more enticements on less number of instalments such as no cost emi.

S.K. Singh