Target SQL Case study

About Target company:

Target is a globally renowned brand and a prominent retailer in the United States. 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 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.

Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

1. Importing the dataset and checking the structure & characteristics of the dataset

1.1. Data Types of Columns

→ BigQuery (BQ):

```
SELECT column_name, data_type

FROM `Target_Brazil.INFORMATION_SCHEMA.COLUMNS`
```

BigQuery Results (BQR):

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

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
8	seller_id	STRING
9	seller_zip_code_prefix	INT64
10	seller_city	STRING
11	seller_state	STRING
12	geolocation_zip_code_prefix	INT64

Using this query we can get an overview of all the columns that are there in the dataset. We can also verify that the data types of columns are what we expect them to be.

1.2. Time period between which the orders were placed.

➔

BQ:

```
SELECT
MIN(order_purchase_timestamp) AS min_date,
MAX(order_purchase_timestamp) AS max_date

FROM `Target_Brazil.orders`
```

BQR:

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

Here from min_date and max_date we can notice the period for which the data is given.

1.3. Cities and States of customers during given year: 2017.

➔ BQ:

```
SELECT
  c.customer_city,
  c.customer_state,
  COUNT(*) AS city_state_count
FROM `Target_Brazil.customers` AS c

INNER JOIN `Target_Brazil.orders` o
ON c.customer_id = o.customer_id

WHERE
  o.order_purchase_timestamp BETWEEN TIMESTAMP('2017-1-1 00:00:00') AND
TIMESTAMP('2017-12-31 00:00:00')
GROUP BY
  c.customer_city,
  c.customer_state

ORDER BY city_state_count DESC
```

BQR:

Row	customer_city	customer_state	city_state_count
1	sao paulo	SP	6367
2	rio de janeiro	RJ	3341
3	belo horizonte	MG	1203
4	brasilia	DF	911
5	porto alegre	RS	691
6	curitiba	PR	650
7	campinas	SP	617
8	salvador	BA	567
9	guarulhos	SP	492

The query shows top9 city-states of the customers who placed an order between 1-1-17 and 31-12-17.

As we can notice, most of the customers are from Sao Paulo followed by Rio, belo horizonte, etc.

2. In-depth Exploration:

2.1. Seasonality trends by months:

➔ BQ:

```
SELECT
  EXTRACT (MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(o.order_id) AS num_of_orders,
  ROUND(SUM(p.payment_value), 0) AS sales_revenue,
  ROUND(SUM(p.payment_value)/ COUNT(o.order_id) , 2) AS sale_per_order

FROM `Target_Brazil.orders` AS o
INNER JOIN `Target_Brazil.payments` AS p
ON o.order_id = p.order_id

WHERE
  o.order_purchase_timestamp BETWEEN TIMESTAMP('2017-01-
01') AND TIMESTAMP('2017-12-31')
GROUP BY month
ORDER BY sales_revenue DESC
```

BQR:

Row	month	num_of_orders	sales_revenue	sale_per_order
1	11	7863	1194883.0	151.96
2	12	5817	868211.0	149.25
3	10	4860	779678.0	160.43
4	9	4516	727762.0	161.15
5	8	4550	674396.0	148.22
6	5	3944	592919.0	150.33
7	7	4317	592383.0	137.22
8	6	3436	511276.0	148.8
9	3	2837	449864.0	158.57
10	4	2571	417788.0	162.5
11	2	1886	291908.0	154.78
12	1	850	138488.0	162.93

As we could observe from the results, number of orders and sales revenue is on uptrend as we approach later part of the year 2017. But it is interesting to see that for the month of January, even though num_orders and sales are lowest, the sale_per_order is highest. This might indicate the domination of high purchase value items or lower discounts compared to other months.

2.2. What time do Brazilians tend to buy?

➔ BQ:

```
SELECT
  CASE
    WHEN (EXTRACT(HOUR FROM o.order_purchase_timestamp) - 3) BETWEEN 0
AND 6
      THEN 'Dawn'
    WHEN (EXTRACT(HOUR FROM order_purchase_timestamp) - 3) BETWEEN 7
AND 12
      THEN 'Morning'
    WHEN (EXTRACT(HOUR FROM order_purchase_timestamp) - 3) BETWEEN 13
AND 18
      THEN 'Afternoon'
    ELSE 'Night'
  END AS time_of_day,
  COUNT(DISTINCT o.order_id) AS num_of_orders
FROM `Target_Brazil.orders` AS o

GROUP BY time_of_day
ORDER BY num_of_orders DESC
```

BQR:

Row	time_of_day	num_of_orders
1	Morning	38291
2	Afternoon	36986
3	Night	14013
4	Dawn	10151

Here, as Brazil time is UTC-03, we had to subtract 3 from the extracted hour. From the results, it is clear that Brazilians are more likely to buy (place the order) in the Morning followed by Afternoon.

3. Evolution of E-commerce orders in the Brazil region:
   3.1. Month on month orders by states:

➔ BQ:

```
SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  c.customer_state AS state,
  COUNT(o.order_id) AS num_of_orders
FROM
  `Target_Brazil.orders`  AS o
JOIN
  `Target_Brazil.customers` AS c
ON
  o.customer_id = c.customer_id
GROUP BY
  year, month, state
ORDER BY
  state, num_of_orders DESC, year, month;
```

BQR:

Row	year	month	state	num_of_orders
1	2017	5	AC	8
2	2017	10	AC	6
3	2018	1	AC	6
4	2017	4	AC	5
5	2017	7	AC	5
6	2017	9	AC	5
7	2017	11	AC	5
8	2017	12	AC	5
9	2017	6	AC	4

This query shows count of orders for each month, each year for respective state. This information can be useful in analyzing order patterns, identifying peak months and understanding customer behavior based on location.

With above query, we can also find the Best and worst month for each state on the basis of no. of orders.

3.2. Distribution of customers across the states in Brazil

→ BQ:

```
SELECT
  customer_state,
  COUNT(customer_unique_id) AS num_of_customers

FROM
  `Target_Brazil.customers`

GROUP BY
  customer_state
ORDER BY
  num_of_customers DESC;
```

BQR:

Row	customer_state	num_of_customers
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

As can be observed, customer_state named SP has highest number of customers and is way ahead of other states in terms of number of customers. SP, RJ, MG are the top3 states based on number of customers.

Below query will give us the total number of customers in Brazil

```
SELECT
  customer_state,
  COUNT(customer_unique_id) AS num_of_customers

FROM
  `Target_Brazil.customers`
```

Result:

Row	num_of_customers
1	99441

So there are total of 99441 customers associated with Target-Brazil.

4. Impact on Economy:
   4.1. % cost increase between 2017 and 2018

→ BQ:

```
SELECT
  (SUM(CASE
        WHEN EXTRACT(Year FROM o.order_purchase_timestamp) = 2018 THEN
p.payment_value
      END
      ) - SUM(CASE
        WHEN EXTRACT(Year FROM o.order_purchase_timestamp) = 2017 THEN
p.payment_value
      END
      ))/ SUM(CASE
      WHEN EXTRACT(Year FROM o.order_purchase_timestamp) = 2017 THEN
p.payment_value
  END
    )*100 AS percent_increase
FROM `Target_Brazil.payments` AS p

JOIN `Target_Brazil.orders` AS o
ON p.order_id = o.order_id

WHERE EXTRACT(Month FROM o.order_purchase_timestamp) BETWEEN 1 AND 8 ;
```

BQR:

Row	percent_increase
1	136.9768716466...

So there is around 137% increase in the revenue from year 2017 and 2018 (between Jan to Aug)

4.2. Mean and Sum of item Price and Freight based on state

➔ BQ:

```
SELECT
  c.customer_state,
  ROUND(AVG(oi.price), 1) AS avg_price,
  ROUND(SUM(oi.price), 1) AS sum_price,
  ROUND(AVG(oi.freight_value), 1) AS avg_freight,
  ROUND(SUM(oi.freight_value), 1) AS sum_freight
FROM
  `Target_Brazil.customers` AS c
  JOIN `Target_Brazil.orders` AS o ON c.customer_id = o.customer_id
  JOIN `Target_Brazil.order_items` AS oi ON o.order_id = oi.order_id

GROUP BY c.customer_state
ORDER BY sum_price DESC
```

BQR:

Row	customer_state	avg_price	sum_price	avg_freight	sum_freight
1	SP	109.7	5202955.1	15.1	718723.1
2	RJ	125.1	1824092.7	21.0	305589.3
3	MG	120.7	1585308.0	20.6	270853.5
4	RS	120.3	750304.0	21.7	135522.7
5	PR	119.0	683083.8	20.5	117851.7
6	SC	124.7	520553.3	21.5	89660.3
7	BA	134.6	511350.0	26.4	100156.7
8	DF	125.8	302603.9	21.0	50625.5
9	GO	126.3	294591.9	22.8	53115.0
10	ES	121.9	275037.3	22.1	49764.6

This information can be useful in logistic management and strategic positioning of warehouses.

From the results, we noticed that the State SP has the lowest avg freight rate while state RR has the highest avg freight rate.

5. Analysis on sales, freight and delivery time

5.1. Calculating the delivery time, the difference between actual delivery date and estimated delivery date

➔ BQ:

```
SELECT
  order_id,

  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) AS days_to_deliver,
  DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,
DAY) AS estimated_delivery_days,
  DATE_DIFF(order_delivered_customer_date,
order_estimated_delivery_date, DAY) AS late_by_days,
  CASE
    WHEN DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY) IS NULL
    THEN "yet_to_be_delivered"
    ELSE "Delivered"
  END AS delivery_status,
FROM `Target_Brazil.orders`

ORDER BY delivery_status DESC;
```

BQR:

Row	order_id	days_to_deliver	estimated_delivery_days	late_by_days	delivery_status
1	7a4df5d8cff4090e541401a20a...	null	16	null	yet_to_be_delivered
2	35de4050331c6c644cddc86f4...	null	33	null	yet_to_be_delivered
3	b5359909123fa03c50bdb0cfe...	null	36	null	yet_to_be_delivered
4	dba5062fbda3af4fb6c33b1e04...	null	25	null	yet_to_be_delivered
5	90ab3e7d52544ec7bc3363c82...	null	24	null	yet_to_be_delivered
6	fa65dad1b0e818e3ccc5cb0e3...	null	27	null	yet_to_be_delivered
7	1df2775799eecdf9dd8502425...	null	31	null	yet_to_be_delivered
8	6190a94657e1012983a274b8...	null	33	null	yet_to_be_delivered
9	58ce513a55c740a3a81e8c8b7...	null	15	null	yet_to_be_delivered
10	088683f795a3d30bfd61152c4f...	null	31	null	yet_to_be_delivered

From this data, we can find out and differentiate delivered and not-delivered orders.

Row	order_id	days_to_deliver	estimated_delivery_days	late_by_days	delivery_status
1	1b3190b2dfa9d789e1f14c05b…	208	19	188	Delivered
2	ca07593549f1816d26a572e06…	209	28	181	Delivered
3	47b40429ed8cce3aee9199792…	191	15	175	Delivered
4	2fe324febf907e3ea3f2aa9650…	189	22	167	Delivered
5	285ab9426d6982034523a855f…	194	28	166	Delivered
6	440d0d17af552815d15a9e41a…	195	30	165	Delivered
7	c27815f7e3dd0b926b5855262…	187	25	162	Delivered
8	0f4519c5f1c541ddec9f21b3bd…	194	32	161	Delivered
9	d24e8541128cea179a11a6517…	175	13	161	Delivered
10	2d7561026d542c8dbd8f0daea…	188	28	159	Delivered
11	6e82dcfb5eada6283dba34f16…	182	27	155	Delivered
12	2fb597c2f772eca01b1f5c561b…	194	39	155	Delivered

Late delivery often creates negative sentiments about the retailer. Using above query, we can also find out the punctuality of the respective orders and can take corrective actions to reduce the fraction of late deliveries in future.

5.2. Analyzing delivery time, freight value by states

➔ BQ:

```
SELECT
  c.customer_state AS state,
  ROUND(AVG(oi.freight_value), 1) AS mean_freight_value,
  ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)), 1) AS mean_time_to_delivery,
  ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_estimated_delivery_date, DAY)), 1) AS mean_late_by_days
FROM `Target_Brazil.customers` AS c

INNER JOIN `Target_Brazil.orders` AS o ON c.customer_id =
o.customer_id
INNER JOIN `Target_Brazil.order_items` AS oi ON o.order_id =
oi.order_id

GROUP BY c.customer_state
ORDER BY 4 DESC  --sort the result by fourth column
```

BQR:

Row	state	mean_freight_value	mean_time_to_delive	mean_late_by_days
1	AL	35.8	24.0	-8.0
2	MA	38.3	21.2	-9.1
3	SE	36.7	21.0	-9.2
4	ES	22.1	15.2	-9.8
5	BA	26.4	18.8	-10.1
6	CE	32.7	20.5	-10.3
7	SP	15.1	8.3	-10.3
8	MS	23.4	15.1	-10.3
9	SC	21.5	14.5	-10.7

Top 5 states with the highest average freight value:

Row	state	mean_freight_value	mean_time_to_delivery	mean_late_by_days
1	RR	43.0	27.8	-17.4
2	PB	42.7	20.1	-12.2
3	RO	41.1	19.3	-19.1
4	AC	40.1	20.3	-20.0
5	PI	39.1	18.9	-10.7

Bottom 5 states according to average freight value:

23	RJ	21.0	14.7	-11.1
24	DF	21.0	12.5	-11.3
25	MG	20.6	11.5	-12.4
26	PR	20.5	11.5	-12.5
27	SP	15.1	8.3	-10.3

Top 5 states with the highest average delivery time:

Row	state	mean_freight_value	mean_time_to_delivery	mean_late_by_days
1	RR	43.0	27.8	-17.4
2	AP	34.0	27.8	-17.4
3	AM	33.2	26.0	-19.0
4	AL	35.8	24.0	-8.0
5	PA	35.8	23.3	-13.4

Bottom 5 states according to average delivery time:

23	SC	21.5	14.5	-10.7
24	DF	21.0	12.5	-11.3
25	MG	20.6	11.5	-12.4
26	PR	20.5	11.5	-12.5
27	SP	15.1	8.3	-10.3

Top 5 states where the order delivery is really fast as compared to the estimated date of delivery:

Row	state	mean_freight_value	mean_time_to_delivery	mean_late_by_days
1	AC	40.1	20.3	-20.0
2	RO	41.1	19.3	-19.1
3	AM	33.2	26.0	-19.0
4	RR	43.0	27.8	-17.4
5	AP	34.0	27.8	-17.4

6. Payment Analysis:
   6.1. Month on Month order count per payment method:

➔ BQ:

```
SELECT
  FORMAT_DATE('%Y-%m', order_purchase_timestamp) AS year_month,
  p.payment_type,
  COUNT(DISTINCT o.order_id) AS order_count
FROM `Target_Brazil.orders` AS o
JOIN `Target_Brazil.payments` AS p
ON o.order_id = p.order_id

GROUP BY year_month,p.payment_type
ORDER BY year_month,p.payment_type
```

BQR:

Row	year_month	payment_type	order_count
1	2016-09	credit_card	3
2	2016-10	UPI	63
3	2016-10	credit_card	253
4	2016-10	debit_card	2
5	2016-10	voucher	11
6	2016-12	credit_card	1
7	2017-01	UPI	197
8	2017-01	credit_card	582
9	2017-01	debit_card	9
10	2017-01	voucher	33
11	2017-02	UPI	398

6.2. Count of orders based on number of payment installments:

➔ BQ:

```
SELECT
  DISTINCT p.payment_installments AS num_pay_inst,
  COUNT(DISTINCT o.order_id) AS order_count
FROM `Target_Brazil.orders` AS o
JOIN `Target_Brazil.payments` AS p
ON o.order_id = p.order_id

GROUP BY 1
ORDER BY 1;
```

BQR:

Row	num_pay_inst	order_count
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644

6.3. Average payment value based on payment type:

➔ BQ:

```
SELECT
  p.payment_type,
  ROUND(AVG(p.payment_value), 1) AS avg_payment_value

FROM `Target_Brazil.payments` AS p
JOIN `Target_Brazil.orders` AS o
ON p.order_id = o.order_id

GROUP BY p.payment_type
ORDER BY p.payment_type
```

BQR:

Row	payment_type	avg_payment_value
1	UPI	145.0
2	credit_card	163.3
3	debit_card	142.6
4	not_defined	0.0
5	voucher	65.7

Here, not_defined in payment_type column might have generated for failed/canceled payments. Average payment value is highest for credit cards.

All these payment insights can help us know how the consumers are spending w.r.t. different payment channels. We can also utilize these payment insights for possible collaboration with payment partners like credit card providers, BNPL firms, Banks, payment apps, etc.

7. Key Actionable Insights:

➔ The general **sales revenue trend is on rise as we go from January to December** month. But interestingly, average sale per order value is more for first half of the year.

➔ Brazilian Customers usually tend to buy in the morning and are least likely to buy in the dawn.

➔ There are total **99441 number of customers** available in the dataset.

➔ There are total **27 states** of Brazil in given dataset.

➔ 80% of those customers are from 6 states (20% of total states) which are SP, RJ, MG, RS, PR, SC.

➔ State **SP alone contribute** to more than **41% of customers,** it also has **lowest average freight value.**

➔ **Number of customers** in a particular state and **average freight** value seems to have **negative correlation.**

➔ Customers in **state PB** seems to order **big ticket items** and has **one of** the **highest average freight value**. This imply that items ordered by customers in PB state are somewhat bulkier.

➔ Payment is **dominated by Credit cards and UPI** methods. Average payment value is highest for credit cards.

➔ Most of the orders belongs to 'single payment installment' type.

8. Key Recommendations:

➔ Company **shall improve the logistics network** in the states where population is lower. Target shall improve its **efficiency** in logistics **network integration**.

➔ There is around 137% increase in revenue from 2017 to 2018 (Jan to Aug), so there is good growth opportunity and it is recommended to Target to stay invested in Brazil market.

➔ Target shall insure that **higher average freight value** in some states is **reflected in timely delivery** of items.

➔ To boosts the revenue, Target shall collaborate with NBFC, BNPL firms, credit card providers to offer attractive **payment schemes or promotions in off season**.

➔ Target shall run **Happy Hours at night time** to have evenly distributed sales and to **avoid excessive traffic in morning and afternoon**.