Business Case: Target SQL

This business case has data of 100k orders from 2016 to 2018 made at Target, Brazil. It is America's leading retailer business chain. Data is available in 8 tables, which gives information about orders from different dimensions like status of order, payment details, location and time of the order , customer who made the purchase, items in the order, product details, seller information of the products, order reviews etc.

1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1.1 Data type of all columns in the "customers" table.

```
SELECT
 column_name,
 data_type
FROM
 `scaler-1stproject-dsml.Target_SQL.INFORMATION_SCHEMA.COLUMNS`
WHERE
 table_name = 'customers';
```

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

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

```
SELECT
 MIN(DATE(order_purchase_timestamp)) AS min_year,
```

```
 MAX(DATE(order_purchase_timestamp)) AS max_year
FROM
 Target_SQL.orders
```

Row	min_year	max_year
1	2016-09-04	2018-10-17

1.3Count the Cities & States of customers who ordered during the given period.

```
SELECT
 DISTINCT c.customer_state,
 c.customer_city
FROM
 `Target_SQL.orders` o
JOIN
 `Target_SQL.customers` c
ON
 o.customer_id=c.customer_id
```

Row	customer_state	customer_city
1	RJ	rio de janeiro
2	RS	sao leopoldo
3	SP	general salgado
4	DF	brasilia
5	PR	paranavai
6	MT	cuiaba
7	MA	sao luis
8	AL	maceio
9	SP	hortolandia
10	MT	varzea grande
11	MG	belo horizonte
12	SP	sao paulo
13	PE	ipojuca
14	SP	itanhaem
15	RS	porto alegre
16	PE	sao lourenco da mata
17	SE	aracaju
18	SP	ituverava

2. In-depth Exploration

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

```
SELECT
 year,
 COUNT(order_id) order_count
FROM (
 SELECT
   *,
   EXTRACT(year
   FROM
     order_purchase_timestamp) AS year
 FROM
   `Target_SQL.orders` )
GROUP BY
 year
ORDER BY
 year
```

Row	year	order_count
1	2016	329
2	2017	45101
3	2018	54011

2.2Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
SELECT
 month,
 COUNT(order_id) order_count
FROM (
```

```sql
SELECT
  *,
  EXTRACT(month
  FROM
    order_purchase_timestamp) AS month
FROM
  `Target_SQL.orders` )
GROUP BY
 month
ORDER BY
 month
```

Row	month	order_count
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

2.3 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

```
WITH
 cte AS(
 SELECT
   order_id,
   CASE
     WHEN TIME(order_purchase_timestamp) BETWEEN "00:00:00" AND "07:00:00" THEN "Dawn"
     WHEN TIME(order_purchase_timestamp) BETWEEN "07:00:01" AND "12:00:00" THEN
"Morning"
     WHEN TIME(order_purchase_timestamp) BETWEEN "12:00:01" AND "18:00:00" THEN
"Afternoon"
     WHEN TIME(order_purchase_timestamp) BETWEEN "18:00:01" AND "23:59:59" THEN
"Night"
 END
   AS purchase_time
 FROM
   `Target_SQL.orders` )
SELECT
 purchase_time,
 COUNT(order_id) AS orer_count
FROM
 cte
GROUP BY
 purchase_time
```

Row	purchase_time	orer_count
1	Morning	21738
2	Dawn	5242
3	Afternoon	38365
4	Night	34096

3.Evolution of E-commerce orders in the Brazil region:
3.1 Get the month on month no. of orders placed in each state.

```
WITH
 cte1 AS(
 SELECT o.order_id,  o.order_purchase_timestamp,
   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
   EXTRACT(Month FROM order_purchase_timestamp) AS month,
   FORMAT_DATE('%b %Y', DATE(ORDER_PURCHASE_TIMESTAMP)) AS time_period,
   c.customer_state AS state
 FROM
   `Target_SQL.orders` o
 JOIN
   `Target_SQL.customers` c
 USING
   (customer_id)
 ORDER BY year, month),

 cte2 AS(
 SELECT state, time_period, year, month, COUNT(*) AS total_orders
 FROM cte1
 GROUP BY state, time_period, year, month )

SELECT state, time_period, total_orders,
 LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month ) AS
prev_month_orders_count,
 ROUND(((total_orders - LAG(total_orders) OVER(PARTITION BY state ORDER BY year,
 month )) / LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month))* 100,2)
AS MoM_percent_growth
FROM
 cte2
ORDER BY
 state
```

Row	state	time_period	total_orders	prev_month_orders_(	MoM_percent_growt
1	AC	Jan 2017	2	*null*	*null*
2	AC	Feb 2017	3	2	50.0
3	AC	Mar 2017	2	3	-33.33
4	AC	Apr 2017	5	2	150.0
5	AC	May 2017	8	5	60.0
6	AC	Jun 2017	4	8	-50.0
7	AC	Jul 2017	5	4	25.0
8	AC	Aug 2017	4	5	-20.0
9	AC	Sep 2017	5	4	25.0
10	AC	Oct 2017	6	5	20.0
11	AC	Nov 2017	5	6	-16.67
12	AC	Dec 2017	5	5	0.0
13	AC	Jan 2018	6	5	20.0
14	AC	Feb 2018	3	6	-50.0
15	AC	Mar 2018	2	3	-33.33
16	AC	Apr 2018	4	2	100.0
17	AC	May 2018	2	4	-50.0
18	AC	Jun 2018	3	2	50.0

3.2 How are the customers distributed across all the states?

```
SELECT customer_state, COUNT(customer_id) custmer_count
FROM
 `Target_SQL.customers`
GROUP BY customer_state
ORDER BY customer_state
```

Row	customer_state	custmer_count
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
11	MG	11635
12	MS	715
13	MT	907
14	PA	975
15	PB	536
16	PE	1652
17	PI	495
18	PR	5045

4.Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
4.1 Get the % increase in the cost of orders from 2017 to 2018 (include months between Jan to Aug only).

```
WITH cte AS(
 SELECT
   o.order_id,
   o.order_purchase_timestamp,
   EXTRACT(year FROM order_purchase_timestamp) AS year,
   EXTRACT(month FROM order_purchase_timestamp) AS month,
   p.payment_value AS payment_value
 FROM
   `Target_SQL.orders` o
 JOIN
   `Target_SQL.payments` p
 USING (order_id)
 WHERE o.order_status='delivered'),
 cte1 AS(
 SELECT year, ROUND(SUM(payment_value),2) AS total_orders_value
 FROM cte
 WHERE year IN (2017, 2018)
   AND month BETWEEN 1 AND 8
 GROUP BY year)
SELECT
 *, COALESCE((ROUND(((total_orders_value - LAG(total_orders_value) OVER(ORDER BY
year))/LAG(total_orders_value) OVER(ORDER BY year))* 100, 2)), 0) AS
percent_increase_YOY
FROM cte1
```

Row	year	total_orders_value	percent_increase_YOY
1	2018	8452975.2	143.33
2	2017	3473862.76	0.0

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

```
SELECT
 c.customer_state,
 ROUND(SUM(oi.price),2) AS Total_price,
 ROUND(AVG(oi.price),2) AS Avg_price
FROM
 `Target_SQL.orders` o
JOIN
 `Target_SQL.order_items` oi
ON
 o.order_id=oi.order_id
JOIN
 `Target_SQL.customers` c
ON
 c.customer_id =o.customer_id
GROUP BY c.customer_state
order by c.customer_state
```

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
11	MG	1585308.03	120.75

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

```
SELECT
 c.customer_state,
 ROUND(SUM(oi.freight_value),2) AS Total_freight,
 ROUND(AVG(oi.freight_value),2) AS Avg_freight
FROM
 `Target_SQL.orders` o
JOIN
 `Target_SQL.order_items` oi
ON
 o.order_id=oi.order_id
JOIN
 `Target_SQL.customers` c
ON
 c.customer_id =o.customer_id
GROUP BY c.customer_state
order by c.customer_state
```

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
11	MG	270853.46	20.63
12	MS	19144.03	23.37
13	MT	29715.43	28.17
14	PA	38699.3	35.83

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

5.1 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.

```
SELECT
 order_id,
 DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS
actual_delivery_time_in_days,
 DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, day) AS
estimated_delivery_time_in_days
FROM
 `Target_SQL.orders`
WHERE
 order_status = "delivered";
```

Row	order_id	actual_delivery_time_in_days	estimated_delivery_time_in_days
1	635c894d068ac37e6e03dc54eccb6189	30	32
2	3b97562c3aee8bdedcb5c2e45a50d5e1	32	33
3	68f47f50f04c4cb6774570cfde3a9aa7	29	31
4	276e9ec344d3bf029ff83a161c6b3ce9	43	39
5	54e1a3c2b97fb0809da548a59f64c813	40	36
6	fd04fa4105ee8045f6a0139ca5b49f27	37	35
7	302bb8109d097a9fc6e9cefc5917d1f3	33	28
8	66057d37308e787052a32828cd007e58	38	32
9	19135c945c554eebfd7576c733d5ebdd	36	33
10	4493e45e7ca1084efcd38ddebf174dda	34	33
11	70c77e51e0f179d75a64a614135afb6a	42	31
12	d7918e406132d7c81f1b845276b03a3b	35	31
13	43f6604e77ce6433e7d68dd86db73b45	32	25
14	37073d851c3f30deebe598e5a586bdbd	31	22
15	d064d4d070d914984df25775004fce96	29	28
16	61d430273ff1e88f2944acb53e99eab5	30	30
17	d2f8ef9dd1714fcac7de9f0aef13d21a	30	21
18	81279a15416799e6580df60f66760a7b	31	18
19	c429654419aacfe84ec52dd4c45f064d	36	17
20	3f6da1442aba80bcf61179602dfab9ca	33	27

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

Highest freight value

```
WITH
 cte AS (
 SELECT
   c.customer_state,
   ROUND(AVG(oi.freight_value),2) AS Avg_freight
 FROM
   `Target_SQL.orders` o
 JOIN
   `Target_SQL.order_items` oi
 ON
   o.order_id=oi.order_id
 JOIN
   `Target_SQL.customers` c
 ON
   c.customer_id =o.customer_id
 GROUP BY c.customer_state
 ORDER BY c.customer_state)
SELECT
 customer_state
FROM cte
ORDER BY Avg_freight DESC
LIMIT 5
```

Row	customer_state
1	RR
2	PB
3	RO
4	AC
5	PI

lowest average freight value.

```
WITH
 cte AS (
 SELECT
   c.customer_state,
   ROUND(AVG(oi.freight_value),2) AS Avg_freight
 FROM
   `Target_SQL.orders` o
 JOIN
   `Target_SQL.order_items` oi
 ON
   o.order_id=oi.order_id
 JOIN
   `Target_SQL.customers` c
 ON
   c.customer_id =o.customer_id
 GROUP BY c.customer_state
 ORDER BY c.customer_state)
SELECT
 customer_state
FROM cte
ORDER BY Avg_freight
LIMIT 5
```

Row	customer_state
1	SP
2	PR
3	MG
4	RJ
5	DF

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

highest average delivery time

```
WITH
 cte AS (
 SELECT
   customer_state,
   TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS
time_to_delivery,
 FROM
   `Target_SQL.orders` o
 JOIN
   `Target_SQL.customers` c
 USING (customer_id)
 WHERE order_status = "delivered" ),
 cte2 AS(
 SELECT
   customer_state,
   ROUND(AVG(time_to_delivery),2) AS avg_delivery_time
 FROM cte
 GROUP BY customer_state )
SELECT customer_state
FROM cte2
ORDER BY avg_delivery_time DESC
LIMIT 5
```

Row	customer_state
1	RR
2	AP
3	AM
4	AL
5	PA

lowest average delivery time

```
WITH
 cte AS (
 SELECT
   customer_state,
   TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS
time_to_delivery,
 FROM
   `Target_SQL.orders` o
 JOIN
   `Target_SQL.customers` c
 USING (customer_id)
 WHERE order_status = "delivered" ),
 cte2 AS(
 SELECT
   customer_state,
   ROUND(AVG(time_to_delivery),2) AS avg_delivery_time
 FROM cte
 GROUP BY customer_state )
SELECT customer_state
FROM cte2
ORDER BY avg_delivery_time
LIMIT 5
```

Row	customer_state
1	SP
2	PR
3	MG
4	DF
5	SC

5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```
WITH
 cte AS (
 SELECT customer_state,
   TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS
time_to_delivery,
   TIMESTAMP_DIFF(order_estimated_delivery_date,order_purchase_timestamp,day) AS
estimate_time_to_delivery
 FROM `Target_SQL.orders` o
 JOIN `Target_SQL.customers` c
 USING (customer_id)
 WHERE order_status = "delivered" ),
 cte2 AS(
 SELECT
   customer_state,
   ROUND(AVG(estimate_time_to_delivery-time_to_delivery),2) AS date_diff
 FROM cte
 GROUP BY customer_state )
SELECT customer_state
FROM cte2
ORDER BY date_diff desc
LIMIT 5
```

Row	customer_state
1	AC
2	RO
3	AP
4	AM
5	RR

6. Analysis based on the payments:

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

```
WITH
 cte AS (
 SELECT
   p.order_id,
   p.payment_type,
   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
   EXTRACT(Month FROM order_purchase_timestamp) AS month,
   FORMAT_DATE('%b %Y', DATE(ORDER_PURCHASE_TIMESTAMP)) AS time_period
 FROM
   `Target_SQL.payments` p
 JOIN
   `Target_SQL.orders` o
 USING
   (order_id) )
SELECT time_period, payment_type, COUNT(*) AS total_orders
FROM cte
GROUP BY time_period, payment_type, YEAR, month
ORDER BY YEAR, month ;
```

Row	time_period	payment_type	total_orders
1	Sep 2016	credit_card	3
2	Oct 2016	credit_card	254
3	Oct 2016	voucher	23
4	Oct 2016	debit_card	2
5	Oct 2016	UPI	63
6	Dec 2016	credit_card	1
7	Jan 2017	voucher	61
8	Jan 2017	UPI	197
9	Jan 2017	credit_card	583
10	Jan 2017	debit_card	9
11	Feb 2017	credit_card	1356

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

```
SELECT payment_installments, COUNT(*) AS total_orders
FROM
 `Target_SQL.payments`
GROUP BY payment_installments;
```

Row	payment_installment	total_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328
12	11	23
13	12	133
14	13	16

Actionable Insights

1. Total 609 orders were unavailable and 625 orders were cancelled during the given time period, which makes it to be around 1.2 % of total orders. We can reduce this number by studying the reasons behind order cancellation and items unavailability.
2. We can see how the orders trajectory is showing a very abrupt increase in orders volume within a very short time. Looking at the overall trend, it is seen that business is picking up very fast in Brazil so companies have to be ready with extra workforce. To avoid high risk, it can consider hiring contractual employees.
3. Companies received low ratings for maximum orders in highlighted states; need to study further about the reasons for customer dissatisfaction to such a great extent in these states.

Recommendations

1. As Brazilian customers usually tend to buy in afternoon and night, we can increase staff during this time frame in order to manage the customers' requests, and services better during this time by reducing the workforce in the morning and dawn.
2. We can see, only 3 states contribute maximum volume, and the rest of the states need to be focused on improving the business.
3. Avg delivery time is quite high for most of those states from where the company is receiving quite less volume of orders, detailed study is needed further for checking the other reasons behind such low volume of orders from majority of states. Huge delivery time can be one of the reasons and we need to work on it.