

Target Brazil E-commerce Analysis

Business Case:

Target, a leading retailer, aims to optimize its Brazilian e-commerce operations.
This project analyzes ~100,000 orders from 2016–2018 to uncover patterns in orders,
payments, delivery performance, and customer satisfaction using Google BigQuery SQL.

Tools:

- Google BigQuery
- SQL (BigQuery Standard SQL dialect)

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

1.A: Check data types of all columns in the "customers" table.

Ans:

```
select
  column_name, data_type
from `target.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

Insights:

We see that customer_id, customer_unique_id, customer_city, customer_state are “STRING” Data Type and customer_zip_code_prefix is of “INTEGER” Data Type.

1.B: Get the time range between which the orders were placed.

Ans:

```
select
  min(order_purchase_timestamp) as start_date,
  max(order_purchase_timestamp) as end_date
from `target.orders`;
```

Query results [Save results](#) [Open in](#)

Job information		Results	Chart	JSON	Execution details	Exec
Row	f0_	f1_				
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC				

Results per page: 50 1 - 1 of 1

Insights:

From the data set, we see that the first order was made in 04.09.2016 and last order was made in 17.10.2018.

1.C. Count distinct cities and states in customer table

```
select
  count(distinct customer_city) as unique_cities,
  count(DISTINCT customer_state) as unique_states
from `target.customers`;
```

Row	city_count	state_count
1	4119	27

Insights:

Customers who ordered are from 4119 cities in 27 states.

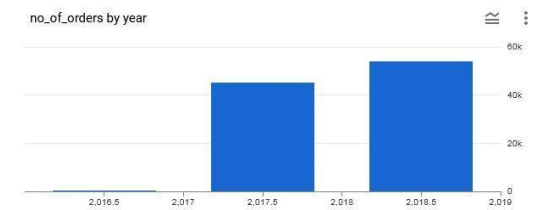
2. In-depth Exploration:

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

Ans:

```
select
  extract(year from order_purchase_timestamp) as year,
  count(*) as no_of_orders
from `target.orders`
group by year
order by year;
```

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



Insights:

Yes, there is a growing trend in the no. of orders placed over the past years.

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

Ans:

```
select month_name, month, year, count(month) as no_of_orders
from ( select *, extract(month from order_purchase_timestamp) as month,
  extract(year from order_purchase_timestamp) as year,
  format_datetime('%b', order_purchase_timestamp) as month_name, from `target.orders`)a
group by 1,2,3
order by 2,3;
```

Row	month_name ▼	month ▼	year ▼	no_of_orders ▼
1	Jan	1	2017	800
2	Jan	1	2018	7269
3	Feb	2	2017	1780
4	Feb	2	2018	6728
5	Mar	3	2017	2682
6	Mar	3	2018	7211
7	Apr	4	2017	2404
8	Apr	4	2018	6939

Insights:

Yes, we can see some kind of monthly seasonality in “Jan 2018 and March 2018” and “Feb 2018 and April 2018”

2.C. During what time of the day, do the Brazilian customers mostly place their orders?
(Dawn, Morning, Afternoon or Night)

i) 0-6 hrs : Dawn ii) 7-12 hrs : Mornings iii) 13-18 hrs : Afternoon iv) 19-23 hrs : Night

Ans:

```
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'
  WHEN extract(hour from order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
END AS time_of_day,
COUNT(*) AS order_count
FROM `target.orders`
GROUP BY time_of_day
ORDER BY order_count DESC;
```

Row	time_of_day ▼	order_count ▼
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

From the given data set we see Brazilian customers mostly place their orders in Afternoon then Night and Mornings. Very few only place their orders in Dawn.

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

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

Ans:

```
select
  customer_state,
  extract(month from order_purchase_timestamp) as month,
  count(*) as no_of_orders,
from `target.orders`
join `target.customers` using (customer_id)
group by customer_state, month
order by customer_state asc, month asc;
```

Row	customer_state ▼	month ▼	no_of_orders ▼
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9

Insights:

From the given data set we found the month on month orders placed in each state.

3.B. How are the customers distributed across all the states?

Ans:

```
select
  customer_state,
  count(customer_id) as no_of_customers
from `target.customers`
group by customer_state
order by no_of_customers desc;
```

Row	customer_state ▼	no_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

Insights:

From the given data set we found that maximum number of customers from SP state.

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

4.A. 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.

Ans:

```
with base_1 as
(
  select
    *
  from `target.orders`
  inner join `target.payments` using (order_id)
  where extract(year from order_purchase_timestamp) between 2017 and 2018 and
  extract(month from order_purchase_timestamp) between 1 and 8
),
base_2 as
(
  select
    extract(year from order_purchase_timestamp) as year,
    round(sum(payment_value),2) as cost
  from base_1
  group by year
  order by year
),
base_3 as
(
  select
    *, lead(cost) over(order by cost) as cost_next_year
  from base_2
)
select *, (cost_next_year - cost) / cost * 100 as percent_increase
from base_3
```

Row	year ▼	cost ▼	cost_next_year ▼	percent_increase ▼
1	2018	8694733.84	null	null
2	2017	3669022.12	8694733.84	136.9768716466...

Insights:

From the given data set we found that 138% increase in the cost of orders from year 2017 to 2018.

4.B. Total & average order price per state

Ans:

```
select
  customer_state ,
  round(sum(price),2) as total_price,
  round(avg(price),2) as average_price
from `target.customers`
inner join `target.orders` using (customer_id)
inner join `target.order_items` using (order_id)
group by customer_state
order by customer_state;
```

Row	customer_state ▼	total_price ▼	average_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

Insights:

From the given data set we found Total price and Average price of product for each state.

4.C. Total & Average value of order freight for each state.

Ans:

```
select
  customer_state ,
  round(sum(freight_value),2) as total_freight_price,
  round(avg(freight_value),2) as average_freight_price
from `target.customers`
inner join `target.orders` using (customer_id)
inner join `target.order_items` using (order_id)
group by customer_state
order by customer_state;
```

Row	customer_state ▼	total_freight_price ▼	average_freight_price ▼
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

Insights:

From the given data set we found Total Freight Price and Average Freight Price of product for each state.

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

5.A. 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.

Ans:

```
select
  order_id,
  timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
  time_to_deliver,
  timestamp_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as
  diff_estimated_delivery
from `target.orders`
where order_status = 'delivered';
```

Row	order_id	time_to_deliver	diff_estimated_d...
1	bfb0f9bdef84302105ad712db...	54	36
2	98974b076b01553d49ee64679...	43	-6
3	c4b41c36dd589e901f6879f25a...	36	-14
4	d2292ff2201e74c5db154d1b7a...	29	-20
5	95e01270fcb9e986342340010...	30	-19
6	ed8c7b1b3eb256c70ce0c7423...	44	-5
7	5cc475c7c03290048eb2e742c...	68	18
8	6b3ee7697a02619a0ace2b3f0a...	47	-2
9	3b2ca3293a7ce539ea2379d70...	43	-7

Insights:

From the given data set we found Delivered time and Difference in estimated delivery time.

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

Ans:

```
select
(a.customer_state) as highest_avg_freight_state, a.highest_avg_freight_value,
(b.customer_state) as lowest_avg_freight_state,
b.lowest_avg_freight_value
from
(select customer_state, round(avg(freight_value),2) as highest_avg_freight_value ,
row_number() over(order by round(avg(freight_value),2) asc) as rnk from `target.customers` c
inner join `target.orders` ord on ord.customer_id = c.customer_id
inner join `target.order_items` o on o.order_id = ord.order_id
group by 1
order by highest_avg_freight_value desc
limit 5) a
inner join
(select customer_state, round(avg(freight_value),2) as lowest_avg_freight_value, row_number()
over(order by round(avg(freight_value),2) desc) rnk from
`target.customers` c
inner join `target.orders` ord on ord.customer_id = c.customer_id
inner join `target.order_items` o on o.order_id = ord.order_id
group by 1
order by lowest_avg_freight_value
limit 5) b on a.rnk = b.rnk ;
```

Row	highest_avg_freight_state	highest_avg_freight_value	lowest_avg_freight_state	lowest_avg_freight_value
1	RR	42.98	SP	15.15
2	PB	42.72	PR	20.53
3	RO	41.07	MG	20.63
4	AC	40.07	RJ	20.96
5	PI	39.15	DF	21.04

Insights:

From the given data set we found Five Highest Avg Freight State and Five Lowest Avg Freight State.

5.C. Top 5 states with the highest & lowest average delivery time.

Ans:

```
select (a.customer_state) as highest_avg_time_deliver_state, a.highest_average_time_deliver,
(b.customer_state) as lowest_avg_time_deliver_state, b.lowest_average_time_deliver
from
  (select customer_state, round(avg(time_to_deliver),2) as highest_average_time_deliver,
  row_number() over(order by round(avg(time_to_deliver),2) desc) as rnk
  from
    (select customer_state, timestamp_diff(order_delivered_customer_date,
    order_purchase_timestamp, day) as time_to_deliver from `target.customers` c
    inner join `target.orders` ord on ord.customer_id = c.customer_id
    inner join `target.order_items` o on o.order_id = ord.order_id)a
  group by 1
  order by 2 desc
  limit 5)a
inner join
  (select customer_state, round(avg(time_to_deliver),2) as lowest_average_time_deliver,
  row_number() over(order by round(avg(time_to_deliver),2) asc) as rnk
  from
    (select customer_state, timestamp_diff(order_delivered_customer_date,
    order_purchase_timestamp, day) as time_to_deliver
    from `target.customers` c
    inner join `target.orders` ord on ord.customer_id = c.customer_id
    inner join `target.order_items` o on o.order_id = ord.order_id)a
  group by 1
  order by 2
  limit 5)b on a.rnk =b.rnk;
```

Row	highest_avg_time_deliver_state	highest_average_time_deliver	lowest_avg_time_deliver_state	lowest_average_time_deliver
1	RR	27.83	SP	8.26
2	AP	27.75	PR	11.48
3	AM	25.96	MG	11.52
4	AL	23.99	DF	12.5
5	PA	23.3	SC	14.52

Insights:

From the given data set we found Five Highest Avg Time Delivery State and Five Lowest Avg Time Delivery State.

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

Ans:

```
select
  customer_state as state,
  round(avg(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp,
day)),2) as avg_del_time,
  round(avg(timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,
day)),2) as avg_est_del_time
from `target.orders`
inner join `target.customers` using (customer_id)
where order_status = 'delivered'
group by state
order by (avg_del_time - avg_est_del_time)
limit 5;
```

Row	state	avg_del_time	avg_est_del_time
1	AC	20.64	40.73
2	RO	18.91	38.39
3	AP	26.73	45.87
4	AM	25.99	44.92
5	RR	28.98	45.63

Insights:

From the given data set we found Five Fastest Delivery State.

6. Analysis based on the payments:

6.A. Month on month no. of orders placed using different payment types.

Ans:

```
select
  extract ( month from order_purchase_timestamp) as month,
  extract ( year from order_purchase_timestamp) as year,payment_type,
  count(distinct order_id) as total_orders
from `target.orders`
inner join `target.payments` using (order_id)
group by month, year, payment_type
order by month, year;
```

Row	month ▼	year ▼	payment_type ▼	total_orders ▼
1	1	2017	credit_card	582
2	1	2017	UPI	197
3	1	2017	debit_card	9
4	1	2017	voucher	33
5	1	2018	credit_card	5511
6	1	2018	UPI	1518
7	1	2018	voucher	304

Insights:

From the given data set we found most of the purchase was done by Credit card only.

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

Ans:

```
select
  payment_installments,
  count(distinct order_id) as num_orders,
from `target.payments`
where payment_installments >= 1
group by payment_installments
order by payment_installments;
```

Row	payment_installm...	num_orders ▼
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623

Insights:

From the given data set we found the installments that have been paid.