

TARGET STORE BUSINESS CASE STUDY USING SQL AND TABLEAU

Context:

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.

By analysing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

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

The data is available in 8 csv files:

1. customers.csv
2. sellers.csv
3. order_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

The column description for these csv files is given below.

The **customers.csv** contain following features:

Features	Description
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer's location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The **sellers.csv** contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller's location

seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The **order_items.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The **geolocations.csv** contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code
geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The **payments.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

The **orders.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.
order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery
order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products

The **reviews.csv** contain following features:

Features	Description
----------	-------------

review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

The **products.csv** contain following features:

Features	Description
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	Length of the string which specifies the name given to the products ordered
product_description_lenght	Length of the description written for each product ordered on the site
product_photos_qty	Number of photos of each product ordered available on the shopping portal
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

What does 'good' look like?

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

Data type of all columns in the “customers” table

QUERY WRITTEN IN BIGQUERY

```
--Data type of all columns in the “customers” table
select column_name , data_type from target.INFORMATION_SCHEMA.COLUMNS
where table_name = "customers"
```

OUTPUT

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

The customers table contain 5 columns namely customer_id with string datatype , customer_unique_id with string datatype , customer_zip_code_prefix with integer datatype , customer_city and customer_state with string datatype

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

To get the date_range , we need the earliest order_purchased_date and latest one and finally find the number of days between them.

QUERY

```
--Get the time range between which the orders were placed.
select min(extract(HOUR from order_purchase_timestamp)) purchased_from, max(extract(HOUR from order_purchase_timestamp)) purchased_till FROM `target.orders`
```

OUTPUT

Row	purchased_from	purchased_till
1	0	23

INFERENCE

Customers placed orders from 12 midnight to 23:00 Hours .It means customers place orders in 24 hours.

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

QUERY

```
--Count the Cities & States of customers who ordered during the given period
select count(distinct customer_city) as city_count, count(distinct customer_state) as state_count from `target.customers`
```

OUTPUT

Row	city_count	state_count
1	4119	27

Evolution of E-commerce orders in the Brazil region:

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

```
-- 1. Getting the month on month no. of orders placed in each state.

select extract(month from o.order_purchase_timestamp) as month , count(o.order_id) as total_orders , c.customer_state
from `target.orders` o
join
`target.customers` c
on
o.customer_id =c.customer_id
group by c.customer_state , month
order by month ,c.customer_state
```

Output

Row	month	total_orders	customer_state
47	2	31	RN
48	2	25	RO
49	2	7	RR
50	2	473	RS

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

```
--How are the customers distributed across all the states?

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

OUTPUT

Row	total_customers	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA

Impact on Economy: Analyse 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).

```

--1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
with cte1 as (
  select * from `target.orders` o
  join
  `target.payments` p
  on
  o.order_id = p.order_id
  where extract(year from order_purchase_timestamp) between 2017 and 2018 and
  extract(month from order_purchase_timestamp) between 1 and 8
),
cte2 as (
  select extract(year from order_purchase_timestamp) as year , round(sum(payment_value),2) as total_payment from cte1
  group by year
)

select * , round((total_payment - lag(total_payment ,1) over(order by year))*100/lag(total_payment) over(order by year),2) as
percent_inc from cte2
order by cte2.year

```

OUTPUT

JOB INFORMATION		RESULTS	CHART	JSON	E
Row	year ▼	total_payment ▼	percent_inc ▼		
1	2017	3669022.12	null		
2	2018	8694733.84	136.98		

INFERENCE : Output shows that there was 137% increase in the cost of orders from year 2017 to 2018.

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

```

select round(sum(ot.price),2) as total_order_price , round(avg(ot.price),2) as average_order_price , c.customer_state from
`target.order_items` ot
join
`target.orders` o
on
ot.order_id = o.order_id
join
`target.customers` c
on
o.customer_id = c.customer_id
group by c.customer_state

```

OUTPUT

Row	total_order_price ▼	average_order_price ▼	customer_state ▼
1	5202955.05	109.65	SP
2	1824092.67	125.12	RJ
3	683083.76	119.0	PR
4	520553.34	124.65	SC
5	302603.94	125.77	DF

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

```
--1. Calculate the Total & Average value of order freight for each state.
select round(sum(ot.freight_value),2) as total_freight_value , round(avg(ot.freight_value),2) as average_freight_value ,
c.customer_state from `target.order_items` ot
join
`target.orders` o
on
ot.order_id = o.order_id
join
`target.customers` c
on
o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state
```

OUTPUT

JOB INFORMATION		RESULTS	CHART	JSON
Row	total_freight_value	average_freight_valu	customer_state	
1	3686.75	40.07	AC	
2	15914.59	35.84	AL	
3	5478.89	33.21	AM	
4	2788.5	34.01	AP	
5	100156.68	26.36	BA	
6	48351.59	32.71	CE	

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.
- 2.

```
--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.

select distinct order_id ,abs(date_diff(order_purchase_timestamp,order_delivered_customer_date , DAY)) as actual_delivery, abs
(date_diff(order_estimated_delivery_date,order_delivered_customer_date , DAY)) as estimated_delivery from `target.orders`
where order_delivered_customer_date is not null
```

OUTPUT

Row	order_id ▼	actual_delivery ▼	estimated_delivery
1	770d331c84e5b214bd9dc70a1...	7	45
2	dabf2b0e35b423f94618bf965f...	7	44
3	8beb59392e21af5eb9547ae1a...	10	41
4	1a0b31f08d0d7e87935b819ed...	6	29
5	cec8f5f7a13e5ab934a486ec9e...	20	40
6	58527ee4726911bee84a0f42c...	10	48
7	10ed5499d1623638ee810eff1...	28	29
8	818996ea247803ddc123789f2...	9	35
9	d105ccc0cccc1204cdc717d28...	10	41

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

For highest average freight value

```
--1. Find out the top 5 states with the highest & lowest average freight value.
--highest freight value
select round(avg(ot.freight_value),2) as highest_freight_value ,
c.customer_state from `target.order_items` ot
join
`target.orders` o
on
ot.order_id = o.order_id
join
`target.customers` c
on
o.customer_id = c.customer_id
group by c.customer_state
order by highest_freight_value desc
limit 5
```

OUTPUT

Row	highest_freight_value	customer_state ▼
1	42.98	RR
2	42.72	PB
3	41.07	RO
4	40.07	AC
5	39.15	PI

LOWEST AVERAGE FREIGHT VALUE


```
--lowest freight value
select round(avg(ot.freight_value),2) as average_freight_value ,
c.customer_state from `target.order_items` ot
join
`target.orders` o
on
ot.order_id = o.order_id
join
`target.customers` c
on
o.customer_id = c.customer_id
group by c.customer_state
order by highest_freight_value
limit 5
```

OUTPUT

Row	avg_freight	customer_state
1	15.15	SP
2	20.53	PR
3	20.63	MG
4	20.96	RJ
5	21.04	DF

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

```
--Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
select c.customer_state,round(avg(abs(date_diff(o.order_purchase_timestamp,o.order_delivered_customer_date , DAY))),2) as
actual_delivery , round(avg(abs(date_diff(o.order_purchase_timestamp,o.order_estimated_delivery_date , DAY))),2) as
estimated_delivery
from `target.orders` o
join
`target.customers` c
on o.customer_id = c.customer_id
where abs(date_diff(o.order_purchase_timestamp,o.order_delivered_customer_date , DAY)) < abs(date_diff(o.
order_estimated_delivery_date,o.order_delivered_customer_date , DAY))
group by c.customer_state
order by abs(actual_delivery - estimated_delivery)
limit 5
```

Press Alt+F1 for accessit

OUTPUT

Row	customer_state ▼	actual_delivery ▼	estimated_delivery
1	SP	5.69	19.71
2	DF	8.27	25.0
3	MG	8.05	25.05
4	PR	8.07	25.19
5	SC	8.82	26.24

Analysis based on the payments:

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

```
--Analysis based on the payments:
--Find the month on month no. of orders placed using different payment types.

select extract(month from o.order_purchase_timestamp) as month ,p.payment_type , count(o.order_id)as Ttoal_Orders from
`target.payments` p
join
`target.orders` o
on
p.order_id = o.order_id
group by month , p.payment_type
order by month
```

OUTPUT

Row	month ▼	payment_type ▼	Total_Orders ▼
1	1	voucher	477
2	1	credit_card	6103
3	1	debit_card	118
4	1	UPI	1715
5	2	credit_card	6609
6	2	voucher	424
7	2	UPI	1723
8	2	debit_card	82
9	3	voucher	591

- Find the no. of orders placed on the basis of the payment instalments that have been paid.

```
--1. Find the no. of orders placed on the basis of the payment instalments that have been paid.

select payment_installments , count(order_id) as Total_Orders from `target.payments`
group by payment_installments
```

OUTPUT

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

BUSINEES INSIGHTS:

- SP has the least average delivery time. Also SP has the least freight value and RR has highest.
- RR has the highest average delivery time
- Sports items, beddings and bath , household items ,home decors , beauty and health are the most popular product category.
- Insurances, Comfort items , CDs ,Gaming devices and fashion wear are the least popular product category.
- Most customers are based in SP states.
- Most purchase were made ahead of holiday season. Hence , offers can be introduced to attract customers .
- Also , freight value can be decreased and delivery time can be increased so that states with low orders can spike.
- Nearly 50% of the customers belong to state SP.