

Target

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.

The data TARGET contains 8 tables as follows:

- customers
- sellers
- order_items
- geolocation
- payments
- reviews
- orders
- products

The column description for these tables 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

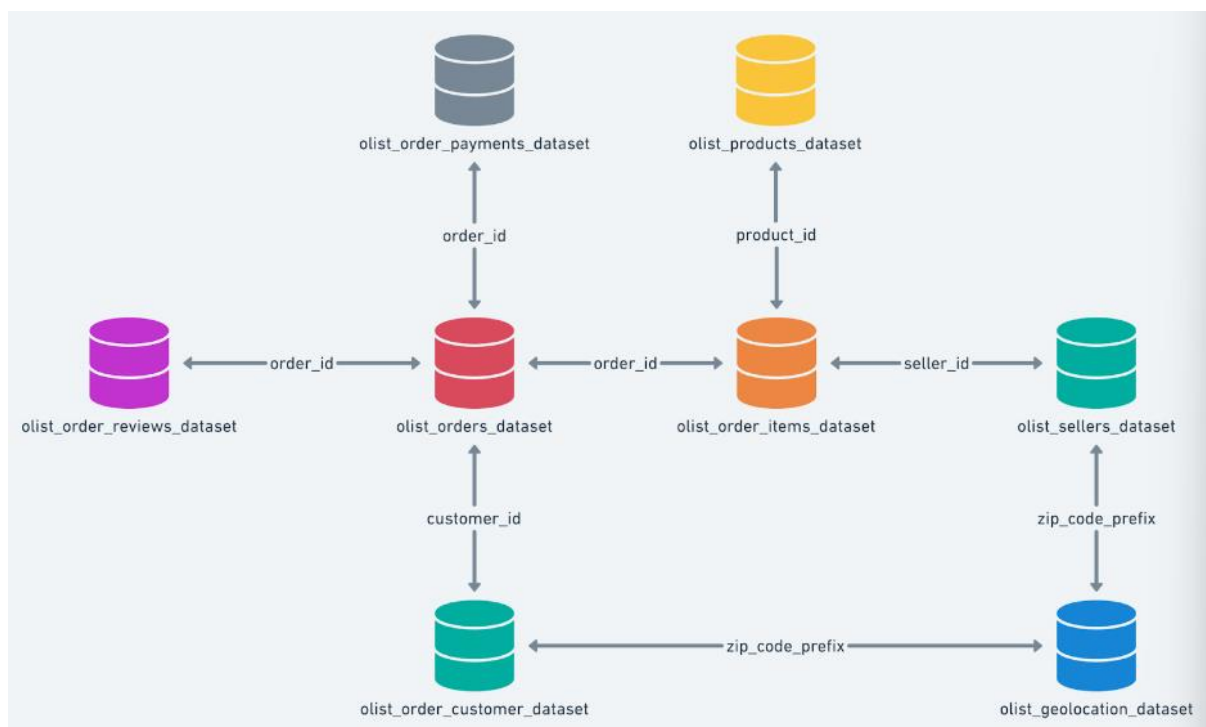
The reviews.csv contain following features:

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

Target Schema:



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

```
SELECT column_name, data_type
FROM `river-direction-384416.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

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

```
select min(order_purchase_timestamp) as dt from `Target.orders`
```

Row	dt ▼
1	2016-09-04 21:15:19 UTC

```
select max(order_purchase_timestamp) as dt from `Target.orders`
```

Row	dt ▼
1	2018-10-17 17:30:18 UTC

All the order were placed between 2016-09-04 21:15:19 and 2018-10-17 17:30:18

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

```
select count(distinct customer_city) from `Target.customers`
```

Row	f0_ ▼
1	4119

```
select count(distinct customer_state) from `Target.customers`
```

Row	f0_ ▼
1	27

Number of distinct cities is 4119 & number of distinct state is 27

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

```
select extract(year from order_purchase_timestamp) as year_,
extract(month from order_purchase_timestamp) as month_,
count (distinct order_id) as num_of_order
from Target.orders a
where order_status='delivered'
group by 1,2
order by 1,2
```

Row	year_ ▼	month_ ▼	num_of_order ▼
1	2016	9	1
2	2016	10	265
3	2016	12	1
4	2017	1	750
5	2017	2	1653
6	2017	3	2546
7	2017	4	2303
8	2017	5	3546
9	2017	6	3135
10	2017	7	3872

We can clearly see that there is increasing trend in the number of orders being placed.

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

```
select mth, count(mth) from
(
select *, extract(month from order_purchase_timestamp) as mth from
`Target.orders`
)
group by mth order by mth asc
```

Row	nth	f0_
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

We can see that customers ordered more in first 8 months from January to August compared to last 4 months with orders peaking in 5th , 7th and 8th month , May , July and August respectively.

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

```

select time_of_day , count(time_of_day) as no_of_orders from
(
select case when hrs between 0 and 6 then 'Dawn'
           when hrs between 7 and 12 then 'Mornings'
           when hrs between 13 and 18 then 'Afternoon'
           when hrs between 19 and 23 then 'Night'
           end as time_of_day
from
(
select *, extract(hour from order_purchase_timestamp) as hrs from
`Target.orders` order by hrs desc

```

```
)
) group by time_of_day
```

Row	time_of_day ▼	no_of_orders ▼
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

We can see that customers have ordered most in the Afternoon and least at dawn.

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

```
select customer_state,
extract(month from order_purchase_timestamp) as month_,
count ( order_id)
from `Target.orders` o join `Target.customers` c on
o.customer_id=c.customer_id
group by 1,2
order by 1,2
```

Row	customer_state ▼	month_ ▼	num_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

- How are the customers distributed across all the states ?

```
select customer_state,
count (o.customer_id) as num_of_customers
from `Target.orders` o join `Target.customers` c on
o.customer_id=c.customer_id
group by 1 order by 2 desc
```


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

9. 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.

```

with cte as
(
select extract( month from order_purchase_timestamp) as mnth , extract (year
from order_purchase_timestamp) as yr ,sum(p.payment_value) as pv from
`Target.payments` p join `Target.orders` o on
o.order_id=p.order_id
group by yr, mnth
having mnth between 1 and 8 and yr between 2017 and 2018
),
cte2 as
(
select yr , sum(pv) as total from cte
group by yr
order by yr asc
), cte3 as (
select yr, total, lead(total) over(order by yr) as next_year from cte2
)
select * , round((((next_year /cte3.total)*100),2) as per_incr from cte3

```

Row	yr	total	next_year	per_incr
1	2017	3669022.120000...	8694733.839999...	236.98
2	2018	8694733.839999...	null	null

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

```
select c.customer_state , round(sum(payment_value),2) as total,
round(avg(payment_value),2) as average from
```

```
`Target.payments` p join `Target.orders` o on o.order_id=p.order_id join
`Target.customers` as c on c.customer_id = o.customer_id
```

```
group by customer_state
```

```
order by customer_state
```

Row	customer_state	total	average
1	AC	19680.62	234.29
2	AL	96962.06	227.08
3	AM	27966.93	181.6
4	AP	16262.8	232.33
5	BA	616645.82	170.82
6	CE	279464.03	199.9
7	DF	355141.08	161.13
8	ES	325967.55	154.71
9	GO	350092.31	165.76
10	MA	152523.02	198.86

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

```
select customer_state, round(sum(freight_value),2) as total ,
round(avg(freight_value),2) as average from
```

```
`Target.customers` c join `Target.orders` o on c.customer_id=o.customer_id
join `Target.order_items` oi on o.order_id=oi.order_id
```

```
group by customer_state
```

Row	customer_state	total	average
1	MT	29715.43	28.17
2	MA	31523.77	38.26
3	AL	15914.59	35.84
4	SP	718723.07	15.15
5	MG	270853.46	20.63
6	PE	59449.66	32.92
7	RJ	305589.31	20.96
8	DF	50625.5	21.04
9	RS	135522.74	21.74
10	SE	14111.47	36.65

12. 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.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date

```
select *, date_diff( extract(date FROM order_delivered_customer_date) ,
extract(date FROM order_purchase_timestamp) , DAY) as time_to_deliver ,
date_diff( extract(date FROM order_delivered_customer_date) , extract(date
FROM order_estimated_delivery_date) , DAY) as diff_estimated_delivery from
`Target.orders` where order_status = 'delivered'
```

Row	time_to_deliver	diff_estimated_delivery
1	31	-2
2	33	-1
3	30	-2
4	44	4
5	41	4
6	37	1
7	34	5
8	39	6
9	36	2
10	34	0

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

```
select seller_state , round(sum(freight_value),2) fr_v from `Target.order_items`
o join `Target.sellers` s on o.seller_id = s.seller_id group by seller_state order by
fr_v limit 5
```

Row	seller_state ▼	fr_v ▼
1	AC	32.84
2	AM	81.8
3	PA	155.11
4	SE	318.49
5	PI	443.32

```
select seller_state , round(sum(freight_value),2) fr_v from `Target.order_items`
o join `Target.sellers` s on o.seller_id = s.seller_id group by seller_state order by
fr_v desc limit 5
```

Row	seller_state ▼	fr_v ▼
1	SP	1482487.67
2	MG	212595.06
3	PR	197013.52
4	SC	106547.06
5	RJ	93829.9

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

```
with cte as
(
select *, date_diff( extract(date FROM order_delivered_customer_date) ,
extract(date FROM order_purchase_timestamp) , DAY) as delivery_time
from `Target.orders` o join `Target.customers` c on o.customer_id
=c.customer_id
where order_status='delivered'
)
select customer_state , round(avg(delivery_time),2) as avg_del_time
from cte group by customer_state order by avg_del_time limit 5
```

Row	customer_state ▼	avg_del_time ▼
1	SP	8.7
2	PR	11.94
3	MG	11.94
4	DF	12.9
5	SC	14.9

```

with cte as
(
select *, date_diff( extract(date FROM order_delivered_customer_date) ,
extract(date FROM order_purchase_timestamp) , DAY) as delivery_time
from `Target.orders` o join `Target.customers` c on o.customer_id
=c.customer_id
where order_status='delivered'
)
select customer_state , round(avg(delivery_time),2) as avg_del_time
from cte group by customer_state order by avg_del_time desc limit 5

```

Row	customer_state	avg_del_time
1	RR	29.34
2	AP	27.18
3	AM	26.36
4	AL	24.5
5	PA	23.73

15. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```

select customer_state , round(avg_of_actual-avg_of_est) as fast from (
select customer_state , round( avg(delivery_days),2) as avg_of_actual ,
round(avg(estimated_delivery_days),2) as avg_of_est from
(
select *, date_diff( extract(date FROM order_delivered_customer_date) ,
extract(date FROM order_purchase_timestamp) , DAY) as delivery_days
,date_diff( extract(date FROM order_delivered_customer_date) ,
extract(date FROM order_estimated_delivery_date) , DAY) as
estimated_delivery_days
from
`Target.orders` o join `Target.customers` c on o.customer_id
=c.customer_id
where order_status='delivered'
) sq
group by customer_state
)
order by fast desc
limit 5

```

Row	customer_state ▼	fast ▼
1	RR	47.0
2	AP	47.0
3	AM	46.0
4	AC	42.0
5	RO	39.0

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

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

Row	payment_type ▼	num_of_payment ▼	mnth ▼
1	voucher	477	1
2	credit_card	6103	1
3	debit_card	118	1
4	UPI	1715	1
5	credit_card	6609	2
6	voucher	424	2
7	UPI	1723	2
8	debit_card	82	2
9	voucher	591	3
10	credit_card	7707	3

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

```
select payment_installments , count(p.order_id ) from `Target.payments`
p join `Target.orders` o on p.order_id=o.order_id group by
payment_installments
```

Row	payment_installment	num_of_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

18. Insights & Recommendation.

- In the Year 2016 data for the month of November is missing and whatever data we have for the year, we can see that the number of orders is too less. What we can infer is that it could be the year of establishment i.e. The company was setting up its store in the country.
- The data for 2018 is only till the month of August, so we have complete year data for year 2017 only. We see that November is the peak month of the year, so the company can maintain the inventory beforehand.
- Any operational challenges should be dealt with, before the peak season of November.
- The company should hire more work force (possibly temporarily) so that the customers don't face any problem during the peak season.
- Most of the customers order most in the afternoon, we can offer them discounts and run advertisements so that customers also order at another time of day.