TARGET BUSINESS CASE STUDY

The data we are going to analyse is from American Retail Company Target Corporation. It was founded on July 1902 and its e-retail stores are very active in American Subcontinent. In our business case, we will focus on whole e-commerce data of Target Brazil from 2016 to 2018. In the dataset it comprises of total 8 tables. The table names are orders, **order\_items, order\_reviews, customer, geolocation, payments, products, sellers**. The datatypes present in our dataset are STRING, INTEGER, FLOAT and TIMESTAMP.

Below are some insights from our case study.

1. Total no. of order placed is **99441**.

2. There are total **99441 customers** in our dataset.

3. Highest payment for a particular order is **13664.08 real**.

4. There are **3095 total sellers** who are associated with Target Brazil, and 1849 from Sao Paolo which is highest is any particular state.

5. RR (Roraima), AL (Alagoas), TO (Tocantins) are three states with **0 sellers**.

6. There are 73 different product category.

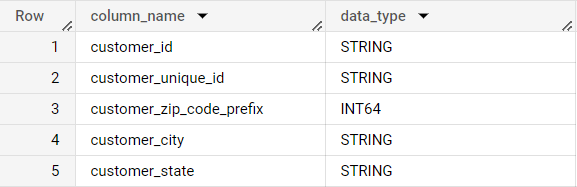
7. Highest number of orders placed in a single day is 24 November 2017 with whooping amount of 1176 orders. The reason behind is **black Friday sale**.

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

SELECT column\_name,data\_type

FROM `target\_sql`.INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name='customers';



There are total 5 columns in a “customers” table with 2 different datatypes which are STRING and FLOAT.

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

SELECT MIN( order\_purchase\_timestamp) minimumpurchasetimestamp,MAX( order\_purchase\_timestamp) maximumpurchasetimestamp

FROM `target\_sql.orders`;

C:\Users\aks75\OneDrive\Pictures\Screenshots\q1.2.png

The earliest order in our dataset is **2016-09-04** and latest is **2018-10-17**.

Q.1.3 Count the number of Cities and States in our dataset.

WITH cte AS(

  SELECT DISTINCT geolocation\_city AS totalcities,

  geolocation\_state AS totalstates

  FROM `target\_sql.geolocation`

  UNION ALL

  SELECT  DISTINCT customer\_city AS totalcities,

   customer\_state AS totalstates

  FROM `target\_sql.customers`

  UNION ALL

  SELECT DISTINCT seller\_city AS totalcities,

   seller\_state AS totalstates

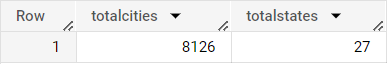
  FROM `target\_sql.sellers`

)

SELECT COUNT( DISTINCT cte.totalcities) as totalcities,

COUNT( DISTINCT cte.totalstates) AS totalstates

FROM cte;



There are total **8126 cities and 27 states** of Brazil in our entire dataset.

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

WITH yearwiseorder AS(

SELECT \*,LAG(ordercount)OVER(ORDER BY orderyear) as previous\_order\_count

FROM

(SELECT

EXTRACT(year FROM order\_purchase\_timestamp) AS orderyear,

COUNT(order\_id) AS ordercount

FROM `target\_sql.orders`

GROUP BY orderyear

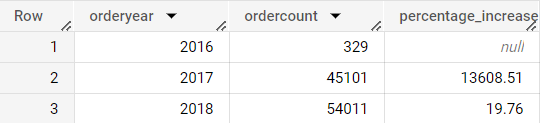
ORDER BY orderyear)tbl)

SELECT orderyear,yearwiseorder.ordercount,

ROUND((ordercount-previous\_order\_count)\*100/previous\_order\_count,2) AS percentage\_increase

FROM yearwiseorder

ORDER BY orderyear;



We can see sudden jump in order counts from 2016 to 2017(around 13609 %) and little more on 2018(around 20%). As in our dataset, only 4 months data is there for 2016. We may suspect that abnormal difference if the whole year data would have been provided.

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

WITH order\_growth\_month\_lead AS

(

SELECT

\*,

LAG(ordercount) OVER(ORDER BY orderyear, ordermonths) AS previous\_month\_order\_count,

FROM

(

SELECT

EXTRACT(year FROM order\_purchase\_timestamp) AS orderyear,

EXTRACT(month FROM order\_purchase\_timestamp) AS ordermonths,

COUNT(order\_id) AS ordercount,

FROM `target\_sql.orders`

GROUP BY ordermonths, orderyear

ORDER BY 1,2

) AS order\_growth\_month

ORDER BY 1,2

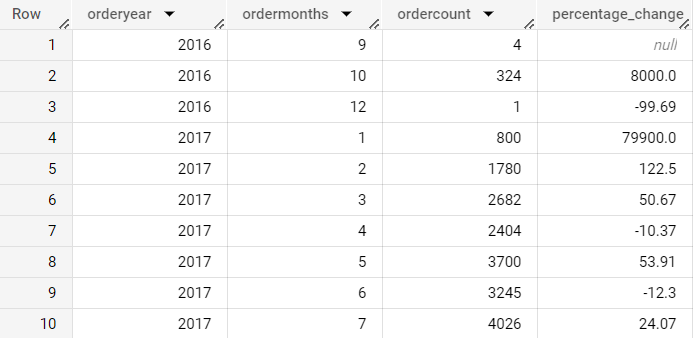
)

SELECT orderyear,ordermonths,ordercount,

ROUND((ordercount-previous\_month\_order\_count)\*100/previous\_month\_order\_count,2) AS percentage\_change

FROM order\_growth\_month\_lead

ORDER BY orderyear,ordermonths;



We can observe that during winter season (from May to September) the order counts are high because it their vacation period and also around October and November because of Black Friday sale.

Q.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 otime AS(SELECT order\_id,

EXTRACT(time FROM order\_purchase\_timestamp) AS ordertime

FROM `target\_sql.orders`)

SELECT COUNT(order\_id) AS ordercount,

CASE

WHEN ordertime BETWEEN '00:00:00' AND '06:59:59'

THEN 'Dawn'

WHEN ordertime BETWEEN '07:00:00' AND '12:59:59'

THEN 'Mornings'

WHEN ordertime BETWEEN '13:00:00' AND '18:59:59'

THEN 'Afternoon'

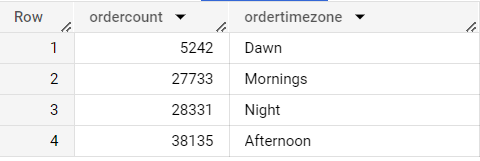
ELSE 'Night'

END AS ordertimezone

FROM otime

GROUP BY ordertimezone

ORDER BY ordercount;



The normal **working hours for Brazil people is 8:00 to 18:00**. Therefore we can study that during night and afternoon order count is high because normally people get free after 18:00 evening.

Q.3.1 Get the month on month no. of orders placed in each state.

SELECT c.customer\_state,

EXTRACT(year FROM order\_purchase\_timestamp) AS orderyear,

EXTRACT(month FROM order\_purchase\_timestamp) AS ordermonth,

COUNT(o.order\_id) AS ordercount

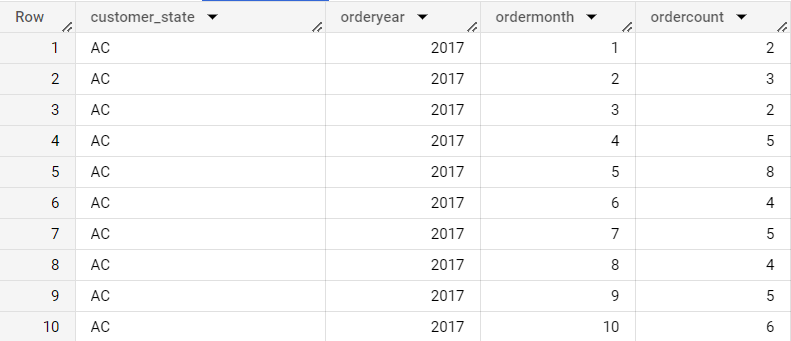
FROM `target\_sql.orders` AS o

INNER JOIN `target\_sql.customers` AS c

ON o.customer\_id = c.customer\_id

GROUP BY ordermonth,c.customer\_state,orderyear

ORDER BY c.customer\_state,orderyear,ordermonth;



Higher number of orders placed during winter and vacation time in Brazil. For example. May, Jun, July and August.

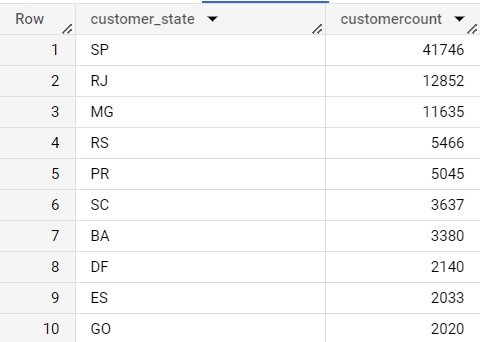
Q.3.2 How are the customers distributed across all the states?

SELECT customer\_state,COUNT(customer\_unique\_id) AS customercount

FROM `target\_sql.customers`

GROUP BY customer\_state

ORDER BY customercount DESC;



Sao Paulo, Rio De Jeneiro, Minas Gerais are **most populated states** of Brazil, therefore they large amount of customers from our dataset.

Q.4.1 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(year from o.order\_purchase\_timestamp) AS year,

  ROUND(SUM(p.payment\_value),2) AS paymentsum

  FROM `target\_sql.orders` AS o

  INNER JOIN `target\_sql.payments` AS p

  ON o.order\_id = p.order\_id

  WHERE EXTRACT(year from o.order\_purchase\_timestamp) IN (2017,2018)

  AND EXTRACT(month from o.order\_purchase\_timestamp) BETWEEN 1 and 8

  GROUP BY year

  ORDER BY year

)

SELECT tb.year,tb.paymentsum,

ROUND(((tb.paymentsum-tb.lag\_value)\*100/tb.lag\_value),2) as growth\_rate

FROM

(SELECT \*,

LAG(paymentsum) over ( order by cte.year asc) as lag\_value

FROM cte

ORDER BY cte.year asc

) as tb;



We can see a significant growth in cost of orders from 2017 to 2018 which is around 137%.

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

SELECT c.customer\_state ,ROUND(SUM(p.payment\_value),2) AS paymentsum,ROUND(AVG(p.payment\_value),2) AS paymentavg

FROM `target\_sql.payments` AS p

INNER JOIN `target\_sql.orders` AS o

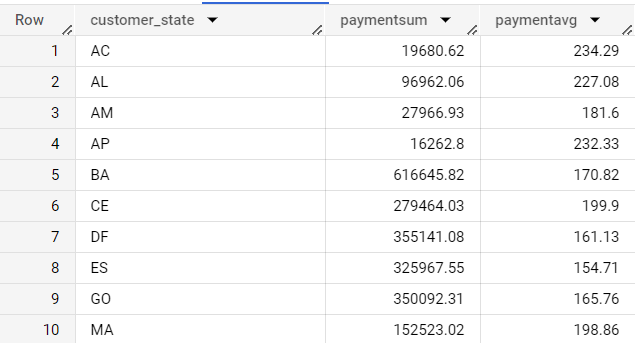
ON p.order\_id=o.order\_id

INNER JOIN `target\_sql.customers` AS c

ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state;



We can observe that payment sum is at least 15000 real and payment average is at least 150 real for all states.

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

SELECT c.customer\_state, ROUND(SUM(oi.freight\_value),2) AS freightsum,ROUND(AVG(oi.freight\_value),2) AS freightavg

FROM `target\_sql.order\_items` AS oi

INNER JOIN `target\_sql.orders` AS o

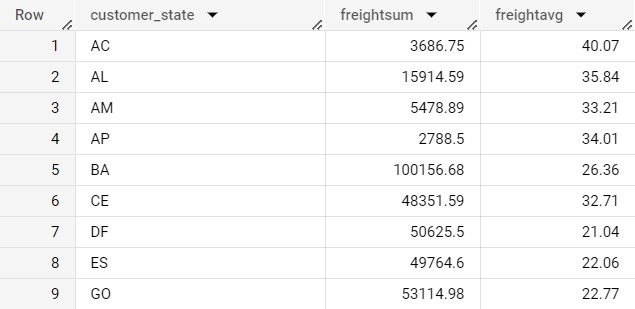
ON oi.order\_id = o.order\_id

INNER JOIN `target\_sql.customers` AS c

ON o.customer\_id=c.customer\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state;



We can observe that freight sum is at least **2700 real** and freight average is at least **21 real** for all states.

Q.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.  
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\_estimated\_delivery\_date - order\_delivered\_customer\_date

SELECT \*

FROM(

SELECT order\_id,order\_purchase\_timestamp,

DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day) AS time\_to\_deliver,

DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day) AS diff\_estimated\_delivery

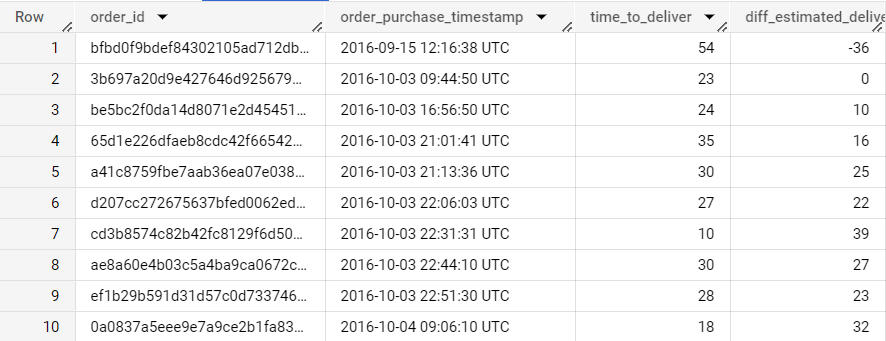
FROM `target\_sql.orders`

ORDER BY order\_purchase\_timestamp

)

WHERE time\_to\_deliver IS NOT NULL

AND diff\_estimated\_delivery IS NOT NULL;



In the above operation we remove those orders which are having null values in order delivery date and estimated delivery date.

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

WITH cte AS(SELECT \*,

DENSE\_RANK()over(ORDER BY avg\_frieght\_value) AS frieghtrank

FROM

(SELECT c.customer\_state,ROUND(AVG(oi.freight\_value),2) AS avg\_frieght\_value

FROM `target\_sql.orders` AS o

INNER JOIN `target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

INNER JOIN `target\_sql.customers` AS c

ON o.customer\_id=c.customer\_id

GROUP BY c.customer\_state

ORDER BY avg\_frieght\_value DESC)tbl

)

SELECT customer\_state,cte.avg\_frieght\_value FROM

cte

WHERE frieghtrank BETWEEN 1 AND 5

OR frieghtrank BETWEEN  23 AND 27

ORDER BY avg\_frieght\_value;



In our table the above 5 rows have **lowest 5 avg\_freight\_value** and below 5 rows have **top 5 highest one**. States like Sao Paulo, Minas Gerais have high demands of orders and distance between seller and customer is less. Which results low freight value. States like RO (Rondonia) and RR (Roraima) are least populated and have less demands. Therefore sellers are very limited. That’s why freight value is higher.

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

WITH cte1 AS(SELECT\*,

ROW\_NUMBER()OVER(ORDER BY time\_to\_deliver) AS timerank

FROM(

SELECT ROUND(AVG(DATE\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,day)),2) AS time\_to\_deliver,

c.customer\_state

FROM `target\_sql.orders` AS o

INNER JOIN `target\_sql.customers` AS c

ON o.customer\_id=c.customer\_id

GROUP BY c.customer\_state

ORDER BY time\_to\_deliver DESC

)tbl

)

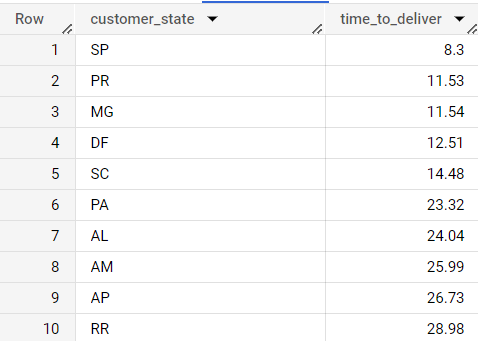
SELECT customer\_state,time\_to\_deliver FROM

cte1

WHERE timerank BETWEEN 1 AND 5

OR timerank BETWEEN 23 AND 27

ORDER BY time\_to\_deliver;



States like (SP) Sao Paulo, (MG) Minas Gerais are **economically powerful** and have **better transportation system**. Therefore it takes less time to reach the customer. On the other hand state like RR (Roraima) are hilly areas and have rough transportation system. Which in result takes more time.

Q.5.4 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 c.customer\_state,ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day)),2) AS time\_to\_deliver

FROM `target\_sql.orders` AS o

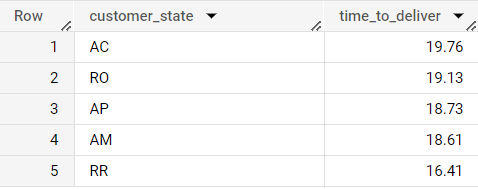
INNER JOIN `target\_sql.customers` AS c

ON o.customer\_id=c.customer\_id

GROUP BY c.customer\_state

ORDER BY time\_to\_deliver DESC

LIMIT 5;



The orders are delivered in these states is less than 20 days.

Q.6.1 Find the month on month no. of orders placed using different payment types.

SELECT EXTRACT(year FROM o.order\_purchase\_timestamp) AS orderyear,

EXTRACT(month FROM o.order\_purchase\_timestamp) AS ordermonth,COUNT(DISTINCT o.order\_id) AS ordercount,p.payment\_type

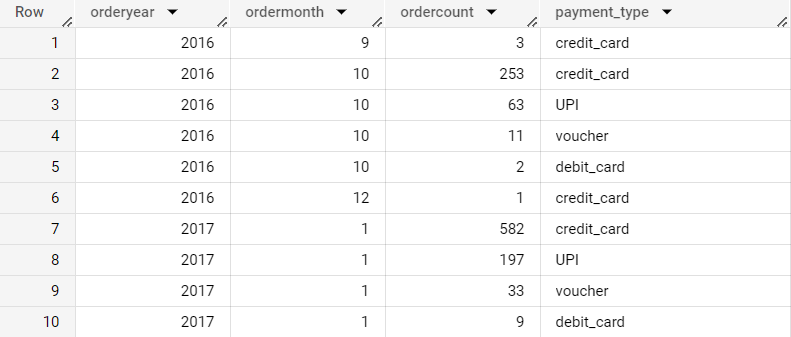
FROM `target\_sql.orders` AS o

INNER JOIN `target\_sql.payments` AS p

ON o.order\_id = p.order\_id

GROUP BY orderyear,ordermonth,p.payment\_type

ORDER BY orderyear,ordermonth;



We can inspect that there is not repetition of usage of particular payment type over the months and years. Customers are using all different type of payments which are available.

Q.6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

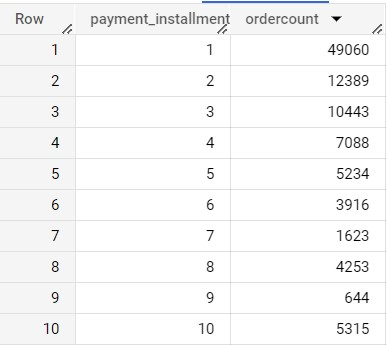
SELECT payment\_installments,COUNT(DISTINCT order\_id) AS ordercount

FROM `target\_sql.payments`

WHERE payment\_installments!=0

GROUP BY payment\_installments

ORDER BY payment\_installments;



We can conclude by the above table that in our dataset majority of the orders have 1, 2 or 3 payment instalment completed.

**RECOMMENDATIONS**

After studying the dataset, we can say that most of the orders, customers, sellers are coming from the major states like Sao Paulo, Rio De Jeneiro or Minas Gerias. Company needs some attention smaller states and improve their services as well. These states also have potential customers, which in results Company may be able to increase their sales and revenue.