

Business Case: 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.

By analyzing 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/1TGEc66YKbD443nsIRi1bWgVd238gJCnb>

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 the 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 The timestamp of the review answered

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

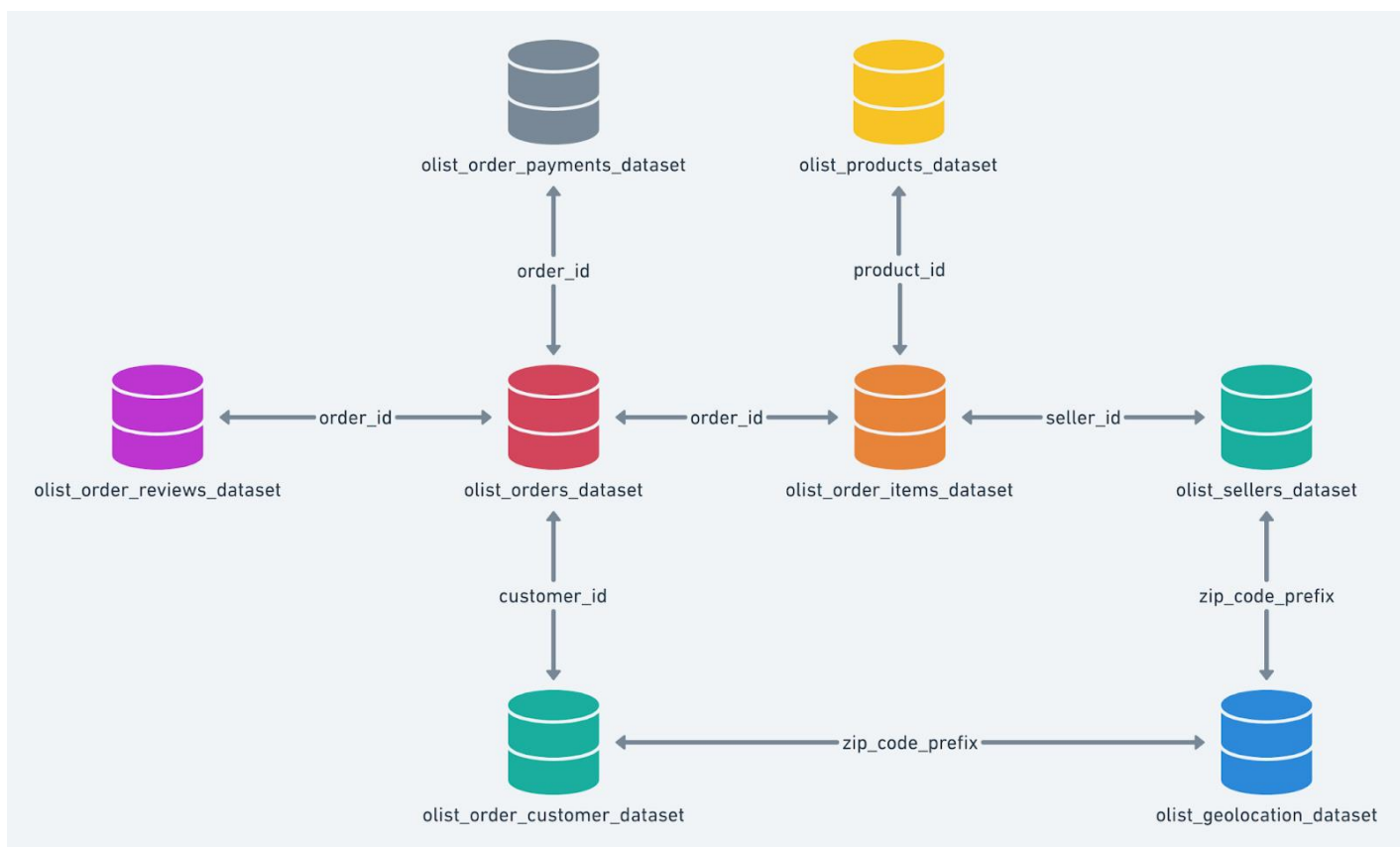
Features

product_id
product_category_name
product_name_lenght
product_description_lenght
product_photos_qty
product_weight_g
product_length_cm
product_height_cm
product_width_cm

Description

A Unique identifier for the proposed project.
Name of the product category
Length of the string which specifies the name given to the products order
Length of the description written for each product ordered on the site
Number of photos of each product ordered available on the shopping por
Weight of the products ordered in grams
Length of the products ordered in centimeters
Height of the products ordered in centimeters
Width of the product ordered in centimeters

Dataset schema:



Problem Statement:

Analyzing the given dataset to extract valuable insights and provide actionable recommendations.

1. *Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:*
 - a. *Data type of all columns in the "customers" table.*
 - b. *Get the time range between which the orders were placed.*
 - c. *Count the Cities & States of customers who ordered during the given period.*

Answer:

1. a)

The data type of all columns in the "customers" table

Result:

Filter Enter property name or value			
<input type="checkbox"/>	Field name	Type	Mode
<input type="checkbox"/>	customer_id	STRING	NULLABLE
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE
<input type="checkbox"/>	customer_city	STRING	NULLABLE
<input type="checkbox"/>	customer_state	STRING	NULLABLE

- ✓ Most of the Column in the Customer Table are in String form and do not contain any Null values.

1.b)

The time range between which the orders were placed

Query:

```
SELECT min(extract(date from order_purchase_timestamp)) as first_purchase_date,
```

```
min(extract(time from order_purchase_timestamp)) as first_purchase_time,
max(extract(date from order_purchase_timestamp)) as last_purchase_date,
max(extract(time from order_purchase_timestamp)) as last_purchase_time
FROM `target-sql-393718.BusinessCase.Orders`
```

Result:

Row	first_purchase_date	first_purchase_time	last_purchase_date	last_purchase_time
1	2016-09-04	00:00:00	2018-10-17	23:59:59

- ✓ The first purchase was done in 2016 on 09/04 at the start of the day and the last purchase was happened in 2018 on 10/17 at end of Day.

1.6

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

Query:

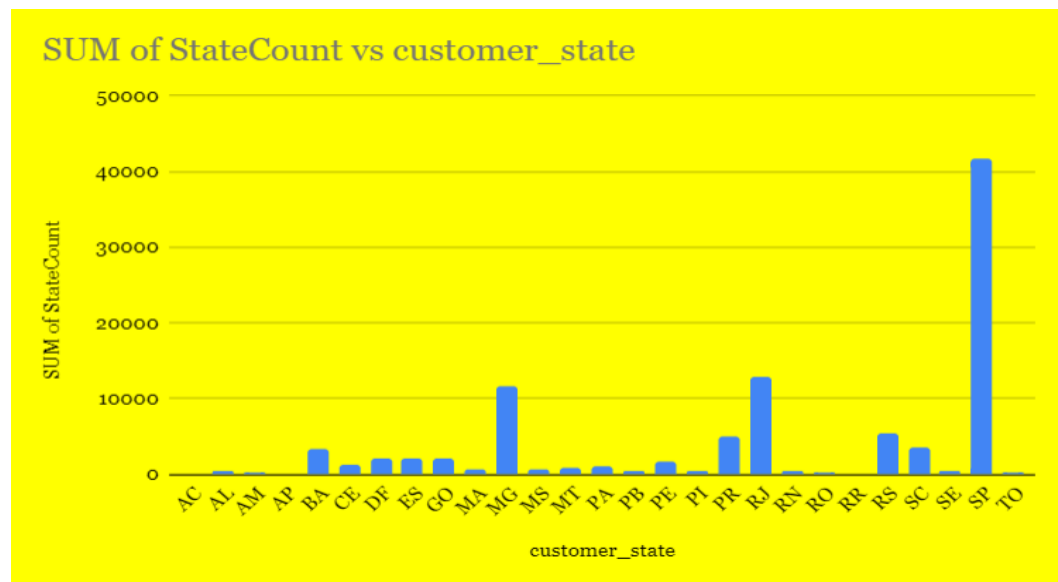
State Count

```
SELECT customer_state, count(*) as StateCount
FROM `target-sql-393718.BusinessCase.Customers`
group by 1
order by StateCount desc
```

Result:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	StateCount		
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		

- ✓ They were a total of 27 States which placed orders during this Duration.
- ✓ The maximum Orders were placed from state SP followed by RJ but there was a huge difference in orders placed in both states and the least orders count were seen to be in State RR having only 46 order counts.



City Count

```
SELECT customer_state, customer_city, count(customer_city) as CityCount
FROM `target-sql-393718.BusinessCase.Customers`
group by 1,2
order by customer_state,CityCount desc
```

Result:

Row	customer_state	customer_city	CityCount
1	AC	rio branco	70
2	AC	cruzeiro do sul	3
3	AC	xapuri	2
4	AC	senador guiomard	2
5	AC	brasileia	1
6	AC	porto acre	1
7	AC	manoel urbano	1
8	AC	epitaciolandia	1
9	AL	maceio	247
10	AL	arapiraca	29
11	AI	nenedo	8

- ✓ They were a total of 4310 Cities that placed orders during this Duration.
- ✓ Most Orders were placed from the City of Sao Paulo having an orders count of 15540 orders from State SP.

2. In-depth Exploration:

- Is there a growing trend in the no. of orders placed over the past years?
- Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
- 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

Answer:

2.a)

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

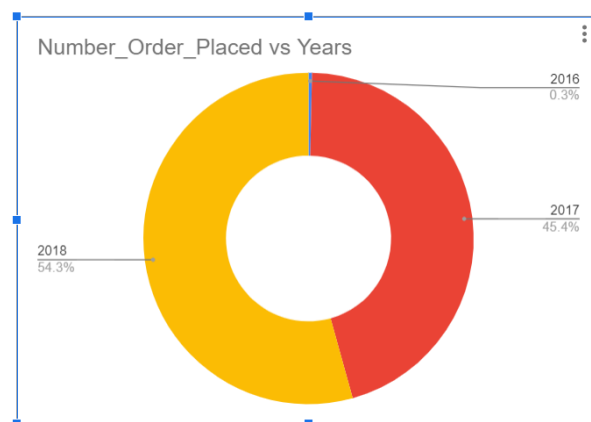
Query:

```
select Years, count(*) as Number_Order_Placed from
(SELECT extract(year from order_purchase_timestamp) as Years
FROM `target-sql-393718.BusinessCase.Orders`
order by Years)as abc
group by Years
order by Years
```

Result:

Row	Years	Number_Order_Placed
1	2016	329
2	2017	45101
3	2018	54011

A	B	C	D	E	F	G	H	I
Years	SUM of Number							
2016	329							
2017	45101							
2018	54011							



- ✓ Yes, the number of orders has significantly increased over the years, It is clearly visible that in 2016 total order count is 329 whereas in 2017 it is 45101 and in 2018 order count went to 54011.

2.b)

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

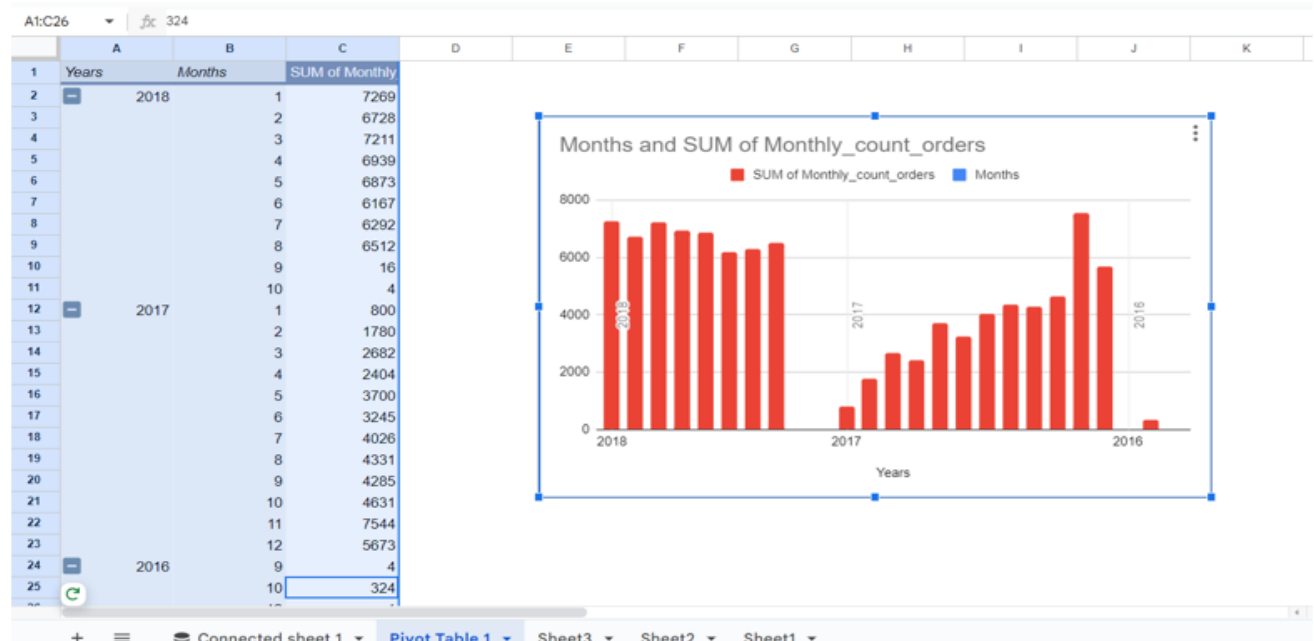
Query:

```
select Years, Months, count(Months) as Monthly_count_orders from
(SELECT extract(year from order_purchase_timestamp) as Years ,extract(month from
order_purchase_timestamp) as Months FROM `target-sql-393718.BusinessCase.Orders`
order by Years, Months)as abc
group by 1,2
order by Years desc, Monthly_count_orders desc
```

Result:

Row	Years	Months	Monthly_count_order
1	2018	1	7269
2	2018	3	7211
3	2018	4	6939

Results per page: 50 1 – 25 of 25



- ✓ In 2016 we can see that major orders were placed in month 10, but in 2017 we can see that as the month moves on the count of orders also move up to 11 months but in 12 months we can see good drops in the Numbers of orders, in 2018 we can see that all months have similar order place pattern and in the month of January maximum orders were placed.

2. c)

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

Query:

```
select Shifts,count(Shifts) as Orders_placed from
(select Hour,
case
when Hour between 0 and 6 then "Dawn"
when Hour between 7 and 12 then "Mornings"
when Hour between 13 and 18 then "Afternoon" else "Night" end as Shifts
from
(SELECT order_purchase_timestamp, extract(hour from order_purchase_timestamp ) as Hour FROM `target-sql-393718.BusinessCase.Orders`
) as abc
) as mno
group by 1
order by Orders_placed desc
```

Result:

Row	Shifts	Orders_placed
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

- ✓ Brazilian customers mostly place their orders in the Afternoon time and then at Night and the least orders are placed at Dawn day time.

3. Evolution of E-commerce orders in the Brazil region:
 - a. Get the month-on-month no. of orders placed in each state.
 - b. How are the customers distributed across all the states?

Answer:

3. a)

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

Query:

```
select Year,Months,customer_state,count(Months) as Total_orders_monthly_state from
(select extract (year from order_purchase_timestamp) as Year ,extract (month from
order_purchase_timestamp) as Months, customer_state from `target-sql-
393718.BusinessCase.Orders` as o join `target-sql-393718.BusinessCase.Customers` as c on
o.customer_id=c.customer_id) as abc
group by 1,2,3
order by Year desc, Total_orders_monthly_state desc
```

Result:

Row	Year	Months	customer_state	Total_orders_monthly_state
1	2018	8	SP	3253
2	2018	5	SP	3207
3	2018	4	SP	3059
4	2018	1	SP	3052
5	2018	3	SP	3037
6	2018	7	SP	2777
7	2018	6	SP	2773
8	2018	2	SP	2703
9	2018	2	RJ	922
10	2018	3	RJ	907

2017

Row	Year	Months	customer_state	Total_orders_monthly_state
223	2017	11	SP	3012
224	2017	12	SP	2357
225	2017	10	SP	1793
226	2017	8	SP	1729
227	2017	9	SP	1638
228	2017	7	SP	1604
229	2017	5	SP	1425
230	2017	6	SP	1331
231	2017	11	RJ	1048
232	2017	3	SP	1010

2016

Row	Year	Months	customer_state	Total_orders_monthly_state
541	2016	10	SP	113
542	2016	10	RJ	56
543	2016	10	MG	40
544	2016	10	RS	24
545	2016	10	PR	19
546	2016	10	SC	11
547	2016	10	GO	9
548	2016	10	CE	8
549	2016	10	PE	7
550	2016	10	DF	6

3.b)

How are the customers distributed across all the states

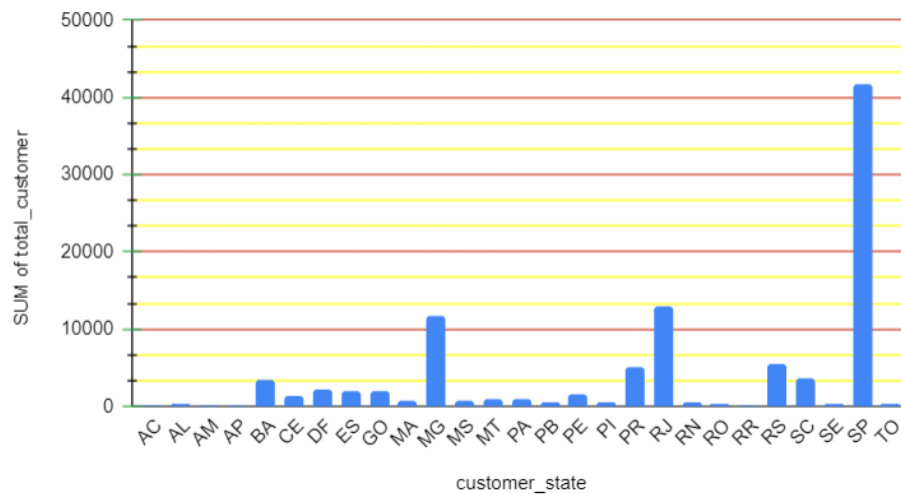
Query:

```
SELECT customer_state,count(customer_id)as total_customer FROM `target-sql-393718.BusinessCase.Customers`  
group by 1  
order by total_customer desc
```

Result:

Row	customer_state	total_customer
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
11	PE	1652

SUM of total_customer



- ✓ **Major of the Target Customer Audience is from SP State as the customer Count in SP is highest and then followed by RJ and MG state.**

4. *Impact on the Economy: Analyze the money movement by e-commerce by looking at order prices, freight, and others.*
- Get the % increase in the cost of orders from the 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.*
 - Calculate the Total & Average value of the order price for each state.*
 - Calculate the Total & Average value of order freight for each state.*

Answer:

4. a)

Get the % increase in the cost of orders from the 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.

Query:

```
with main_1 as (
select month,count(p_2017) as P_2017 from
(SELECT extract(month from order_purchase_timestamp) as month , payment_value as p_2017
FROM `target-sql-393718.BusinessCase.Payments` as p
join `target-sql-393718.BusinessCase.Orders` as o on o.order_id = p.order_id
```

```

where extract(year from order_purchase_timestamp) =2017 and extract(month from
order_purchase_timestamp) in (1,2,3,4,5,6,7,8)) as abc
group by 1
order by month asc)
,
main_2 as (
select month, count(p_2018) as P_2018 from
(SELECT extract(month from order_purchase_timestamp) as month , payment_value as p_2018
FROM `target-sql-393718.BusinessCase.Payments` as p
join `target-sql-393718.BusinessCase.Orders` as o on o.order_id = p.order_id
where extract(year from order_purchase_timestamp) =2018 and extract(month from
order_purchase_timestamp) in (1,2,3,4,5,6,7,8)) as abc
group by 1
order by month asc)

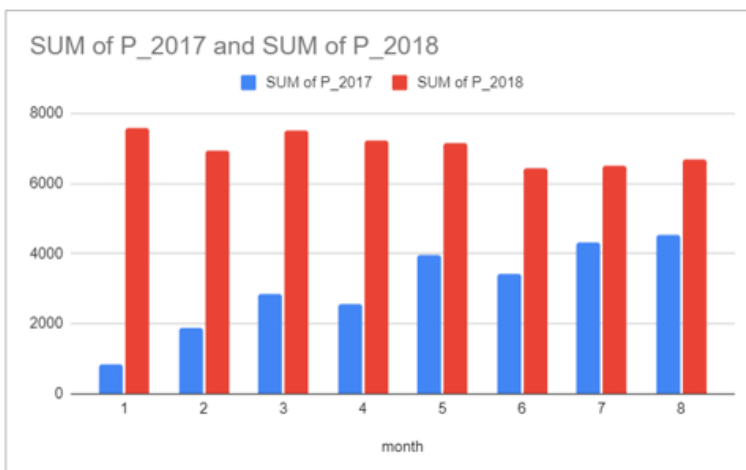
select a1.month,a1.P_2017,a2.P_2018 ,round(((a2.P_2018-a1.P_2017)/a1.P_2017)*100,2) as
percent_increase
from main_1 as a1 join main_2 as a2 on a1.month=a2.month
order by a1.month

```

Result:

Row	month	P_2017	P_2018	percent_increase
1	1	850	7563	789.76
2	2	1886	6952	268.61
3	3	2837	7512	164.79
4	4	2571	7209	180.4
5	5	3944	7135	80.91
6	6	3436	6419	86.82
7	7	4317	6507	50.73
8	8	4550	6698	47.21

month	SUM of P_2017	SUM of P_2018
1	850	7563
2	1886	6952
3	2837	7512
4	2571	7209
5	3944	7135
6	3436	6419
7	4317	6507
8	4550	6698



- ✓ In the Month of January, there was a significant increase in the cost of orders whereas in the month of August we can observe less difference in the cost of order increment.

4.b)

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

Query:

```
select year,customer_state,round(sum(payment_value),2) as total_states_order_price
,round(avg(payment_value),2) as avg_states_order_price from
(SELECT extract(year from order_purchase_timestamp) as year, customer_state, payment_value
FROM `target-sql-393718.BusinessCase.Payments` as p
join `target-sql-393718.BusinessCase.Orders` as o on o.order_id = p.order_id
join `target-sql-393718.BusinessCase.Customers` as c on o.customer_id=c.customer_id)as abc
group by 1,2
order by year asc, total_states_order_price desc
```

Result:

2016

Row	year	customer_state	total_states_order_price	avg_states_order_price
1	2016	SP	16885.54	143.1
2	2016	RJ	13407.58	206.27
3	2016	MG	5642.97	137.63
4	2016	RS	4790.7	184.26
5	2016	SC	2730.42	248.22
6	2016	PR	2599.97	118.18
7	2016	CE	2011.77	251.47
8	2016	PE	1688.49	241.21
9	2016	PA	1283.09	320.77
10	2016	GO	1223.06	122.31
11	2016	DF	1200.11	200.02

2017

Row	year	customer_state	total_states_order_price	avg_states_order_price
22	2017	SP	2561862.91	136.02
23	2017	RJ	1055954.48	160.7
24	2017	MG	854891.64	150.62
25	2017	RS	424693.3	153.43
26	2017	PR	353782.25	149.15
27	2017	BA	284324.38	166.27
28	2017	SC	277452.28	155.78
29	2017	GO	164005.35	164.17
30	2017	DF	157089.83	164.66
31	2017	PE	153281.89	187.62

2018

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	year	customer_state		total_states_order_price	avg_states_order_price
40	2016	MT		45093.21	186.34
41	2017	RN		44650.87	175.1
42	2017	SE		40833.8	205.19
43	2017	RO		31483.72	215.64
44	2017	TO		28827.94	207.4
45	2017	AM		13031.61	164.96
46	2017	AC		12997.52	232.1
47	2017	AP		7331.53	244.38
48	2017	RR		2006.58	111.48
49	2018	SP		3419478.51	138.61
50	2018	RJ		1075017.63	156.0

- ✓ In all years we can observe that Total sales order prices were highest by state SP and then followed by state RJ whereas the Average value of order price in 2016 is highest for state PA and MT and for 2017 this frequency is maintained by state AP, AL and for 2018 it is for state RR and PB

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

- 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:

- $time_to_deliver = order_delivered_customer_date - order_purchase_timestamp$
 - $diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date$
- Find out the top 5 states with the highest & lowest average freight value.
 - Find out the top 5 states with the highest & lowest average delivery time.
 - 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

Answer:

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.

Query:

```
select * from
(SELECT date_diff(order_delivered_customer_date,order_purchase_timestamp, Day) as
time_to_deliver,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,Day) as
diff_estimated_delivery
FROM `target-sql-393718.BusinessCase.Orders`
order by time_to_deliver desc)as abc
where abc.time_to_deliver is not Null
```

Result:

Row	time_to_deliver	diff_estimated_delivery
1	209	181
2	208	188
3	195	165
4	194	161
5	194	166
6	194	155
7	191	175
8	189	167
9	188	159
10	187	144

✓ The time to Deliver can vary between (1-209) and the fastest delivery is the same day

5.2)

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

Top 5 States with highest average freight value

Query:

```
with abc as
(SELECT customer_state,freight_value
FROM `target-sql-393718.BusinessCase.Customers` as c join `target-sql-
393718.BusinessCase.Orders` as o
on c.customer_id=o.customer_id
join `target-sql-393718.BusinessCase.Order_Items` as oi
on o.order_id=oi.order_id)
,
```

```

abc1 as
(select customer_state,round(avg(freight_value),0) as avg_freight_val from abc
group by 1
order by avg_freight_val desc)
,
abc2 as
(SELECT customer_state,avg_freight_val,row_number() over(order by avg_freight_val desc)as
row_number1 from abc1)

select customer_state,avg_freight_val from abc2
where (row_number1>=1 and row_number1<=5)

```

Result:

Row	customer_state	avg_freight_val
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0

✓ The major States having high Freight Value are PB, RR, RO, AC, PI

Top 5 States with lowest average freight value

Query:

```

with abc as
(SELECT customer_state,freight_value
FROM `target-sql-393718.BusinessCase.Customers` as c join `target-sql-393718.BusinessCase.Orders` as o
on c.customer_id=o.customer_id
join `target-sql-393718.BusinessCase.Order_Items` as oi
on o.order_id=oi.order_id)
,
abc1 as
(select customer_state,round(avg(freight_value),0) as avg_freight_val from abc
group by 1
order by avg_freight_val desc)
,
abc2 as
(SELECT customer_state,avg_freight_val,row_number() over(order by avg_freight_val asc)as
row_number1 from abc1)

select customer_state,avg_freight_val from abc2
where (row_number1>=1 and row_number1<=5)

```

Result:

Row	customer_state	avg_freight_val
1	SP	15.0
2	MG	21.0
3	RJ	21.0
4	DF	21.0
5	PR	21.0

✓ The states having low freight value are SP, MG, RJ, DF, PR

5.3)

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

Top 5 States with highest Average Delivery time**Query:**

```
select customer_state from
(SELECT customer_state,average_del_time,row_number() over(order by average_del_time desc)as
row_number1 from
(select customer_state,round(avg(time_to_deliver),0) as average_del_time from
(SELECT customer_state, date_diff(order_delivered_customer_date,order_purchase_timestamp, Day)
as time_to_deliver,
FROM `target-sql-393718.BusinessCase.Orders` as o join `target-sql-
393718.BusinessCase.Customers` as c
on o.customer_id=c.customer_id) as abc
group by 1
order by average_del_time desc) abc1
) as abc2
```

where (row_number1>=1 and row_number1<=5)

Result:

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

✓ State RR, AP, AM, AL, PA are observed to have the highest Delivery time.

Top 5 States with lowest Average Delivery time

Query:

```
select customer_state from
(SELECT customer_state,average_del_time,row_number() over(order by average_del_time asc)as
row_number1 from
(select customer_state,round(avg(time_to_deliver),0) as average_del_time from
(SELECT customer_state, date_diff(order_delivered_customer_date,order_purchase_timestamp, Day)
as time_to_deliver,
FROM `target-sql-393718.BusinessCase.Orders` as o join `target-sql-
393718.BusinessCase.Customers` as c
on o.customer_id=c.customer_id) as abc
group by 1
order by average_del_time desc) abc1
) as abc2

where (row_number1>=1 and row_number1<=5)
```

Result:

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. You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Query:

```
with main as(
SELECT
customer_state,date_diff(order_estimated_delivery_date,order_delivered_customer_date,Day) as
Days
FROM `target-sql-393718.BusinessCase.Orders` as o join `target-sql-
393718.BusinessCase.Customers` as c
on o.customer_id=c.customer_id
where order_delivered_customer_date < order_estimated_delivery_date and
order_delivered_customer_date is not Null)
```

```
select customer_state,round(avg(Days),0)as fast_delivery_avg_days_by_state from main
group by customer_state
order by fast_delivery_avg_days_by_state desc
limit 5
```

Result:

Row	customer_state	fast_delivery_avg_da
1	RR	24.0
2	AP	22.0
3	AC	21.0
4	AM	20.0
5	RO	20.0

✓ State RR, AP, AC, AM, and RO are observed to have the fastest delivery states in Brazil

6. Analysis based on the payments:

1. Find the month-on-month no. of orders placed using different payment types.
2. Find the no. of orders placed on the basis of the payment installments that have been paid.

Answers:

6.1)

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

Query:

```
with main as
(SELECT Extract(year from order_purchase_timestamp) as year, Extract(month from
order_purchase_timestamp) as month ,
case when payment_type='credit_card' then 1 else 0 end as credit,
case when payment_type='UPI' then 1 else 0 end as UPI,
case when payment_type='voucher' then 1 else 0 end as Voucher,
case when payment_type='debit_card' then 1 else 0 end as Debit,
case when payment_type='not_defined' then 1 else 0 end as NotDefine
FROM `target-sql-393718.BusinessCase.Orders` as o join `target-sql-
393718.BusinessCase.Payments` as p
on o.order_id=p.order_id
order by year desc,month asc)

select year,month,sum(credit) as CreditMode,sum(UPI) as UPIMode, sum(Voucher) as VoucherMode,
sum(Debit) as DebitMode,sum(NotDefine) as NDMode
from main
```

group by year,month
order by year desc, month asc

Result:

2018

Row	year ▼	month ▼	CreditMode ▼	UPIMode ▼	VoucherMode ▼	DebitMode ▼	NDMode ▼
1	2018	1	5520	1518	416	109	0
2	2018	2	5253	1325	305	69	0
3	2018	3	5691	1352	391	78	0
4	2018	4	5455	1287	370	97	0
5	2018	5	5497	1263	324	51	0
6	2018	6	4813	1100	324	182	0
7	2018	7	4755	1229	281	242	0
8	2018	8	4985	1139	295	277	2
9	2018	9	0	0	15	0	1
10	2018	10	0	0	4	0	0

2017

Row	year ▼	month ▼	CreditMode ▼	UPIMode ▼	VoucherMode ▼	DebitMode ▼	NDMode ▼
11	2017	1	583	197	61	9	0
12	2017	2	1356	398	119	13	0
13	2017	3	2016	590	200	31	0
14	2017	4	1846	496	202	27	0
15	2017	5	2853	772	289	30	0
16	2017	6	2463	707	239	27	0
17	2017	7	3086	845	364	22	0
18	2017	8	3284	938	294	34	0
19	2017	9	3283	903	287	43	0
20	2017	10	3524	993	291	52	0

2016

23	2016	9	3	0	0	0	0
24	2016	10	254	63	23	2	0
25	2016	12	1	0	0	0	0

✓ Majority of the Payment is done by using credit Mode and UPI Mode.

6.2]

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

Query:

```
select payment_installments, count(order_id) as no_of_order
from `target-sql-393718.BusinessCase.Payments`
group by payment_installments
order by no_of_order desc;
```

Result:

Row	payment_installment	no_of_order
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644
11	12	133
12	15	74
13	18	27
14	11	23
15	24	18
16	20	17
17	13	16
18	14	15

Row	payment_installment	no_of_order
16	20	17
17	13	16
18	14	15
19	17	8
20	16	5
21	21	3
22	0	2
23	22	1
24	23	1

- ✓ Based on fewer Installments Customers have made large purchases on that basis and for installments greater than 14 orders placed are fewer.

Insights and Recommendations:

- 📊 Orders count increased over the years we see in 2016 order count was 329 and in 2017 order count was 45101 which is an excellent performance by Target to achieve such a good Difference in the Count of orders but in 2018 order count increased to 54011 which is good but if we see the Difference between order count of 2017 and 2018 it not much a Good increment as between 2016 and 2017.

- ✚ Target should focus on more product launches, Ads, and Discounts in the Afternoon and Night Day time as most customers usually placed their orders at this time. Also to Increase order count in Dawn Day time Target should see which category of People do not purchase and approach them by giving Heavy deals on Products. Another way is to launch sales offers like a Big Billion sale to launch at Dawn time so that customers will be aware of Dawn Time Sales.
- ✚ In 2018 Major orders were placed from State SP in the months of August, May, April, and January, and 2017 again large orders were placed by State SP in the months of November and December, and 2016 all order counts of the year were done in October month and major are placed from SP and RJ State.
- ✚ Target should focus on States where the total order count in all duration time is less than 200 then states customers were their major focus by giving them extra discount coupons, Fast Delivery, and extra customer care help. The States are AM, AC, AP, and RR.
- ✚ For years 2017 and 2018 Target made a huge profit in the months of January and February if we see aggregately and the month of August is not having a good profitable month.
- ✚ Customers from State SP are seen to have high-paying Jobs or rich and have the lowest average freight and lowest delivery time value for the years 2016-2018 if the order price is summed then they were the highest one.
- ✚ The time to deliver should be not greater than 100 Days, Target should focus on their Delivery Partners.
- ✚ Target should more focus on Payment terms like Debit Mode and Voucher mode which can help to increase performance in Sales as some customers use these modes but due to fewer ties up with the bank, are unable to make purchases successful.
- ✚ The product should have the facility to make payment in installments, usually if the number of installments is less then customers were seen to be happy and make a large number of orders placed,

✚ Also, the target should charge less interest rates on installments as it is seen as the best helpful way to pay money and increase the number of orders.