

# Business case study : TARGET SQL

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

## 1.1 Datatype and column of the table?

Answer 1.1 : We have the following tables along with their columns and Datatypes in our Target case study:

### a) customers table

*# using this query to calculate datatype for all.*

```
SELECT COLUMN_NAME, DATA_TYPE  
FROM Target_SQL_Business_Case.INFORMATION_SCHEMA.COLUMNS  
WHERE TABLE_name = 'customers'
```

Field name	Type
<u>customer_id</u>	STRING
<u>customer_unique_id</u>	STRING
<u>customer_zip_code_prefix</u>	INTEGER
<u>customer_city</u>	STRING
<u>customer_state</u>	STRING

### b) geolocation

Field name	Type
<u>geolocation_zip_code_prefix</u>	INTEGER
<u>geolocation_lat</u>	FLOAT
<u>geolocation_lng</u>	FLOAT
<u>geolocation_city</u>	STRING
<u>geolocation_state</u>	STRING

### c) order\_items

Field name	Type
<u>order_id</u>	STRING
<u>order_item_id</u>	INTEGER
<u>product_id</u>	STRING
<u>seller_id</u>	STRING
<u>shipping_limit_date</u>	TIMESTAMP
<u>price</u>	FLOAT
<u>freight_value</u>	FLOAT

### d) order\_reviews

Field name	Type
<u>review_id</u>	STRING
<u>order_id</u>	STRING
<u>review_score</u>	INTEGER
<u>review_comment_title</u>	STRING
<u>review_creation_date</u>	TIMESTAMP
<u>review_answer_timestamp</u>	TIMESTAMP

e) orders



Field name	Type
<u>order_id</u>	STRING
<u>customer_id</u>	STRING
<u>order_status</u>	STRING
<u>order_purchase_timestamp</u>	TIMESTAMP
<u>order_approved_at</u>	TIMESTAMP
<u>order_delivered_carrier_date</u>	TIMESTAMP
<u>order_delivered_customer_date</u>	TIMESTAMP
<u>order_estimated_delivery_date</u>	TIMESTAMP

f) payments



Field name	Type
<u>order_id</u>	STRING
<u>payment_sequential</u>	INTEGER
<u>payment_type</u>	STRING
<u>payment_installments</u>	INTEGER
<u>payment_value</u>	FLOAT

g) products



Field name	Type
<u>product_id</u>	STRING
<u>product_category</u>	STRING
<u>product_name_length</u>	INTEGER
<u>product_description_length</u>	INTEGER
<u>product_photos_qty</u>	INTEGER
<u>product_weight_g</u>	INTEGER
<u>product_length_cm</u>	INTEGER
<u>product_height_cm</u>	INTEGER
<u>product_width_cm</u>	INTEGER

h) sellers



Field name	Type
<u>seller_id</u>	STRING
<u>seller_zip_code_prefix</u>	INTEGER
<u>seller_city</u>	STRING
<u>seller_state</u>	STRING

## 1.2 Time period for which data is given?

Answer 1.2, Here we will deal with the orders table and will do the following steps:

Step 1: Find out the minimum order date from 'order\_purchase\_timestamp' and we will extract year from this timestamp:

7	
8	SELECT EXTRACT(year FROM MIN(order_purchase_timestamp)) as min_year
9	FROM `Target_SQL_Business_Case.orders`
Query results	
JOB INFORMATION RESULTS JSON EXECUTION DETAILS	
Row	min_year
1	2016

Step 2: Finding the year which is the maximum and difference of max and min year will give us the time period.

```

8 SELECT EXTRACT(YEAR FROM MAX(order_purchase_timestamp)) as max_year,
9 EXTRACT(YEAR FROM MAX(order_purchase_timestamp)) - EXTRACT(YEAR FROM MIN(order_purchase_timestamp)) as time_period
10 FROM `Target_SQL_Business_Case.orders`

```

Press Alt+F1 for accessibility

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

EXECUTION GRAPH

PREVIEW

Row	max_year	time_period
1	2018	2

#### OBSERVATION:

1. We have given data from 2016 till 2018 ie 2 years data is given to us.
2. We used 'orders' table cause it tells us the details cause logically first order will tell us the date on which we incorporated the data and difference between first order and last order will give us the range of time period.

### 1.3 Cities and states of customers ordered during the given period?

Answer 1.3, Here we need to find the customers who ordered the products and we will do the following steps:

Step 1: Logically orders table contains customer ID who ordered product.

Step 2: We will do Inner join or left join with customers table to get Cities and states.

Step 3: Let's assume the given period is the date on which customers ordered the products, we will convert order\_purchase\_timestamp into datetime to get the proper date format.

```
SELECT c.customer_city as City,c.customer_state as State,
FORMAT_TIMESTAMP("%b-%d-%Y",o.order_purchase_timestamp) as Order_date
FROM `Target_SQL_Business_Case.orders` as o
INNER JOIN `Target_SQL_Business_Case.customers` as c ON o.customer_id = c.customer_id
ORDER BY c.customer_city
```



Row	City	State	Order_date
1	abadia dos dourados	MG	Jul-02-2018
2	abadia dos dourados	MG	Sep-02-2017
3	abadia dos dourados	MG	Mar-02-2018
4	abadiania	GO	Jan-28-2018
5	abaete	MG	Nov-23-2017
6	abaete	MG	Jun-01-2018
7	abaete	MG	Jul-15-2017
8	abaete	MG	Aug-31-2017
9	abaete	MG	Jul-31-2017
10	abaete	MG	Mar-27-2018

## 2. In depth exploration.

2.1, Is there a growing trend on e commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

**Answer 2.1, We will do the following steps:**

**Step 1: To find the seasonality we will count the number of orders based on the months, ie first we will convert our timestamp in month and year format.**

**Step 2: Then we will count number of orders that comes under the months and year, we can use GROUP BY or WINDOWS Fn for that.**

**Step 3: We will use matplotlib to plot the data.**

```

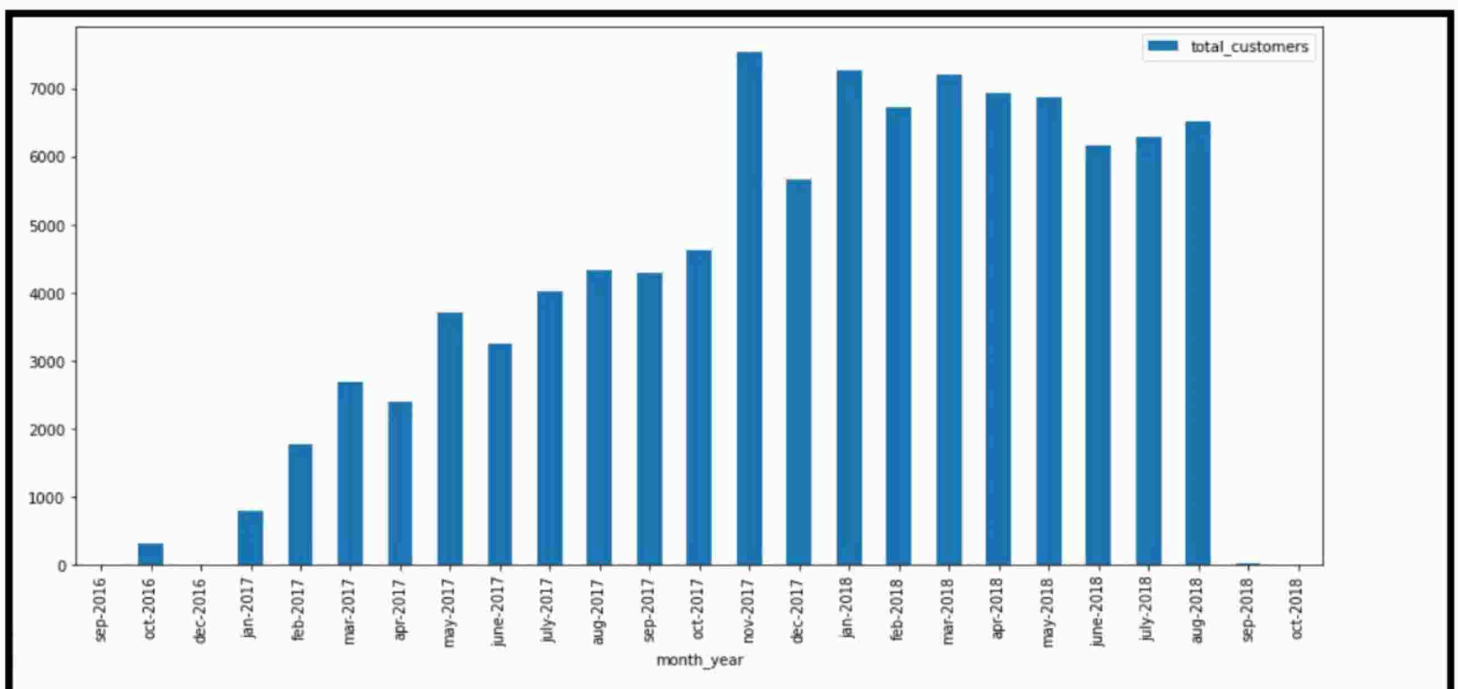
SELECT table1.year,table1.month,COUNT(table1.customer_id) as totat_cust
FROM
  (SELECT customer_id,extract(year from order_purchase_timestamp)as year,
    extract(month from order_purchase_timestamp) as month,
    CAST(order_purchase_timestamp AS DATE) as date1
  FROM `Target_SQL_Business_Case.orders`) as table1
GROUP BY table1.year,table1.month
ORDER BY table1.year,table1.month

```

OUTPUT:

Row	year	month	totat_cust
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728
18	2018	3	7211
19	2018	4	6939
20	2018	5	6873
21	2018	6	6167
22	2018	7	6292
23	2018	8	6512
24	2018	9	16
25	2018	10	4

# Using matplotlib to plot the above data





## OBSERVATION:

1. We have the maximum number of customer orders in the month of November 2017.
2. We can also observe the increasing trend as can conclude that number of customers are increasing as compare to initial months which is amplifying the good performance of the company.

2.2, What time do Brazilian customers tend to buy(Dawn,Morning, Afternoon or Night)?

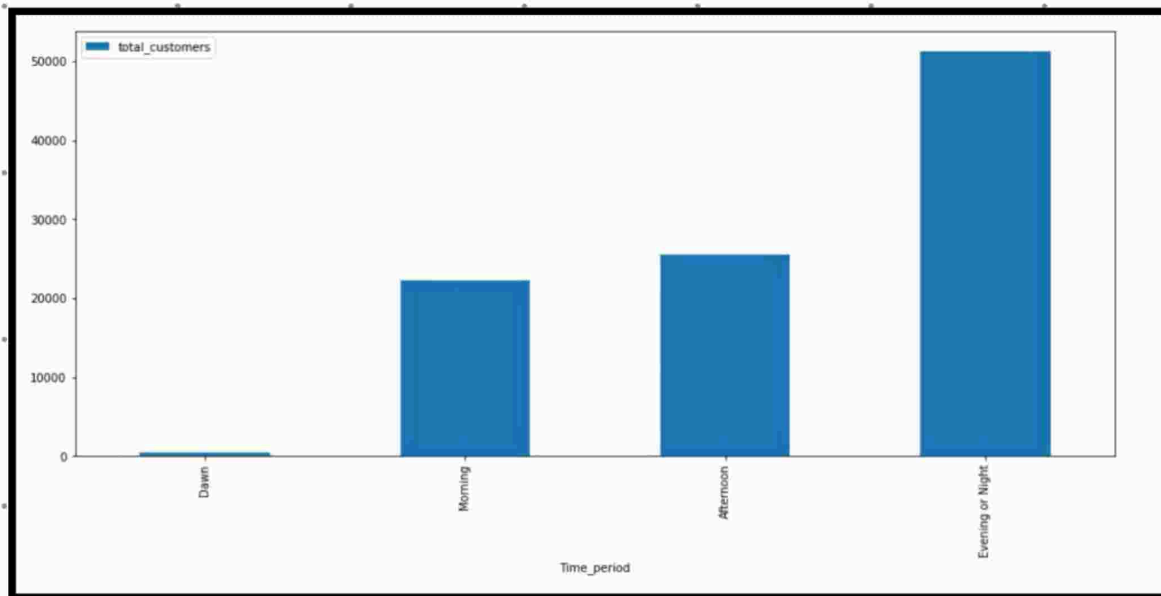
Answer 2.2, We will follow the below steps:

Step 1: To get the details of the time of order we will look into the "order\_purchase\_timestamp" and will extract time from this timestamp.

Step 2: We will convert this timestamp into 4 parts as asked in the question and will plot the graph for it.

```
SELECT t1.time_period,COUNT(t1.customer_id) as Customers
FROM
  (SELECT customer_id,(case when extract(hour from order_purchase_timestamp) >= 4 and
    | | extract(hour from order_purchase_timestamp) < 6
    | | then 'Dawn'
    | | when extract(hour from order_purchase_timestamp) >= 6 and
    | | extract(hour from order_purchase_timestamp) < 12
    | | then 'Morning'
    | | when extract(hour from order_purchase_timestamp) >= 12 and
    | | extract(hour from order_purchase_timestamp) < 16
    | | then 'Afternoon'
    | | else 'Evening or Night'
  end) as time_period
FROM `Target_SQL_Business_Case.orders`) t1
GROUP BY t1.time_period
ORDER BY COUNT(t1.customer_id)
```

Row	time_period	Customers
1	Dawn	394
2	Morning	22240
3	Afternoon	25536
4	Evening or Night	51271



#### OBSERVATION:

1. We can observe that Brazilians tend to buy more in the evening or night time followed by afternoon.
2. This behavior is mostly true for all customers around the world as they are busy in daytime and go for shopping at the evening or night time.
3. During the dawn, very less number of customers are buying compared to other time brackets.

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

#### 3.1, Get month to month orders by states?

Answer 3.1, We will be following the below steps:

Step 1: To get the month to month orders we will use orders table and will use GROUP BY on order\_purchase\_timestamp by extracting month and year.

Step 2: To get the states data we will look into customers table and will do GROUP BY on customer\_state column.

NOTE: Here we are doing two GROUP BY one is on states and other on months/year.

```

SELECT *
FROM
  (SELECT c.customer_state,COUNT(c.customer_id) AS total,
    EXTRACT(month FROM o.order_purchase_timestamp) as Months
  FROM `Target_SQL_Business_Case.orders` o INNER JOIN `Target_SQL_Business_Case.customers` c
  ON o.customer_id = c.customer_id
  GROUP BY c.customer_state,EXTRACT(month FROM o.order_purchase_timestamp)
  ) t1
ORDER BY t1.total desc

```

Row	customer_state	total	Months
1	SP	4982	8
2	SP	4632	5
3	SP	4381	7
4	SP	4104	6
5	SP	4047	3
6	SP	3967	4
7	SP	3357	2
8	SP	3351	1
9	SP	3012	11
10	SP	2357	12
11	SP	1908	10
12	SP	1648	9



## OBSERVATION:

1. We can see that the above state is "SP" has the maximum number of customers.
2. We used Tableau to plot the data we got from the query.



### 3.2, Distribution of customers across the states in Brazil?

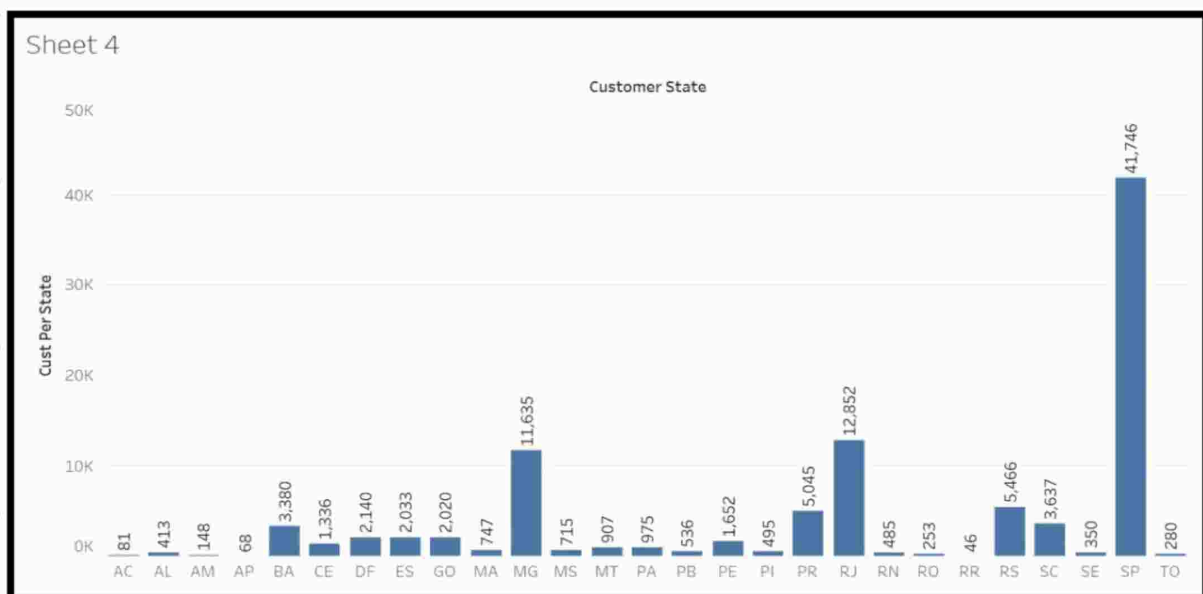
Answer 3.2, We will do the following steps:

Step 1: To get the states we will look into customers table.

Step 2: We will use GROUP BY or window fn to get total number of customers per state.

```
SELECT customer_state ,COUNT(customer_id) as cust_per_state
FROM `Target_SQL_Business_Case.customers`
GROUP BY customer_state
ORDER BY customer_state
```

row	customer_state	cust_per_state
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635
12	MS	715
13	MT	907
14	PA	975
15	PB	536
16	PE	1652
17	PI	495
18	PR	5045
19	RJ	12852
20	RN	485
21	RO	253
22	RR	46
23	RS	5466
24	SC	3637
25	SE	350
26	SP	41746
27	TO	280



#### OBSERVATION:

1. The state "SP" (Sao Paulo) has the maximum number of customers ie. 41746 customers.
2. This state may have the highest population in the Brazil hence having highest number of customers,

#### 4. Impact on economy: Analyze the money movement by E-commerce by looking at the order prices, freight and others.

4.1, Get % increase in the cost of orders from 2017 to 2018(include months between Jan to Aug only)- You can use payment\_value column in the payments table.

Answer 4.1, We will do the following steps:

Step 1: We will extract year and month from orders table from order\_purchase\_timestamp using JOIN and will select year 2017 and 2018, months between Jan and Aug.

Step 2: We use the formula =  $(\text{Current} - \text{Base} / \text{Base}) \times 100$  for the percentage in months from 2017 to 2018.

```
# creating CTE for year 2017
with YEAR_2017 AS
(SELECT t1.year1, t1.month1, ROUND(sum(t1.payment_value)) as payment
FROM
  (SELECT p.payment_value, EXTRACT(year from o.order_purchase_timestamp) as year1, EXTRACT(month from o.order_purchase_timestamp) as month1
   FROM 'Target_SQL_Business_Case.payments' p INNER JOIN 'Target_SQL_Business_Case.orders' o
    ON p.order_id = o.order_id) t1
WHERE t1.year1 in (2017) and t1.month1 BETWEEN 1 AND 8
GROUP BY t1.year1, t1.month1
ORDER BY t1.year1, t1.month1),
# creating CTE for year 2018
YEAR_2018 AS
(SELECT t2.year1, t2.month1, ROUND(sum(t2.payment_value)) as payment
FROM
  (SELECT p.payment_value, EXTRACT(year from o.order_purchase_timestamp) as year1, EXTRACT(month from o.order_purchase_timestamp) as month1
   FROM 'Target_SQL_Business_Case.payments' p INNER JOIN 'Target_SQL_Business_Case.orders' o
    ON p.order_id = o.order_id) t2
WHERE t2.year1 in (2018) and t2.month1 BETWEEN 1 AND 8
GROUP BY t2.year1, t2.month1
ORDER BY t2.year1, t2.month1)

# calculating the % increase = (Curr-Base/Base)*100
SELECT CONCAT(t4.year_2017, '-', t4.year_2018) as years, t4.months_2017 as month,
ROUND(((t4.payment_2018-t4.payment_2017)/t4.payment_2017)*100, 0) as percentage_change #((Curr-Base/Base)*100
FROM
  (SELECT y1.year1 as year_2017, y1.month1 as months_2017, y1.payment as payment_2017,
   y2.year1 as year_2018, y2.month1 as months_2018, y2.payment as payment_2018
   FROM YEAR_2017 y1 INNER JOIN YEAR_2018 y2 on y1.month1 = y2.month1
   ORDER BY y1.year1, y1.month1) t4
```

*performing self join here*

Row	years	month	percentage_change
1	2017-2018	1	705.0
2	2017-2018	2	240.0
3	2017-2018	3	158.0
4	2017-2018	4	178.0
5	2017-2018	5	95.0
6	2017-2018	6	100.0
7	2017-2018	7	80.0
8	2017-2018	8	52.0

#### OBSERVATION:

1. In the month of January we can see there's 705% increase in payments compared Jan 2018 with Jan 2017.
2. The reason there's such a huge increase because in Jan 2017, the company might be having less number of customers or orders compared to Jan 2018 where the company's customers got increased resulting in high payments.

#### 4.2, Mean and Sum of price and freight value by customer state?

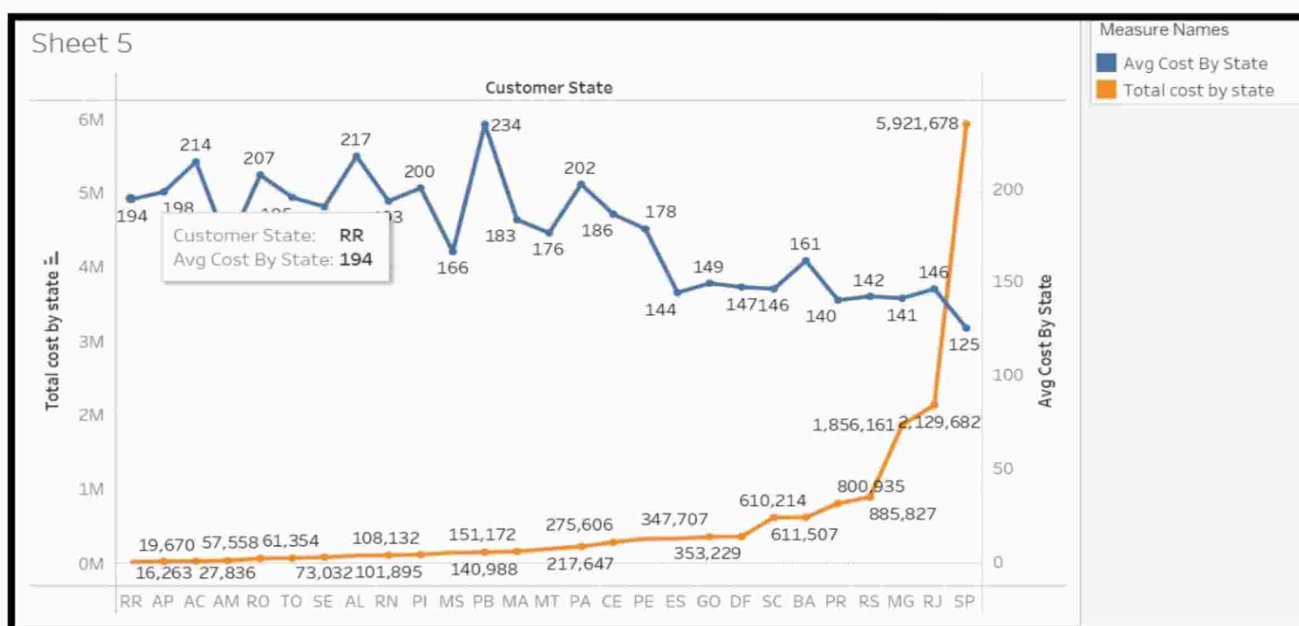
Answer 4.2, We will do the following steps:

Step 1: Information about the price and freight in present in the order\_items table.

Step 2: The customer state information is present in the customers table and we will perform the JOIN on order\_items, customers, orders and then we will do GROUP BY operation.

```
SELECT c.customer_state, ROUND(SUM(ot.price+ot.freight_value)) as Total_cost_by_state,
ROUND(AVG(ot.price+ot.freight_value)) as avg_cost_by_state,
FROM `Target_SQL_Business_Case.order_items` ot JOIN `Target_SQL_Business_Case.orders` o
ON ot.order_id = o.order_id JOIN `Target_SQL_Business_Case.customers` c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Total_cost_by_state
```

Row	customer_state	Total_cost_by	avg_cost_by
1	RR	10065.0	194.0
2	AP	16263.0	198.0
3	AC	19670.0	214.0
4	AM	27836.0	169.0
5	RO	57558.0	207.0
6	TO	61354.0	195.0
7	SE	73032.0	190.0
8	AL	96229.0	217.0
9	RN	101895.0	193.0
10	PI	108132.0	200.0
11	MS	135957.0	166.0
12	PB	140988.0	234.0
13	MA	151172.0	183.0
14	MT	186169.0	176.0
15	PA	217647.0	186.0
16	CE	275606.0	144.0
17	PE	322238.0	149.0
18	ES	324802.0	178.0
19	GO	347707.0	146.0
20	DF	353229.0	142.0
21	SC	610214.0	141.0
22	BA	611507.0	161.0
23	PR	800935.0	140.0
24	RS	885827.0	146.0
25	MG	1856161.0	125.0
26	RJ	2129682.0	125.0
27	SP	5921678.0	125.0



## OBSERVATION:

1. It is interesting to know that states where we have the highest revenue (price+freight) have the lower average revenue. The average measure gives a deep understanding on how the Brazilian states are performing.
2. The states with less revenue have the more average revenue.

## 5. Analysis on sales, freight, and delivery time:

### 5.1, Calculate days between purchasing, delivery, and estimated delivery?

Answer 5.1, We will do the following steps:

Step 1: The purchasing , delivery , and estimated delivery data is in the orders table.

```
SELECT t.customer_id,t.order_id,
IFNULL(DATE_DIFF(t.delivery_date,t.purchasing_date,DAY),0)AS diff_purchasing_delivery,#difference b/w delivery and purchasing date
IFNULL(DATE_DIFF(t.estimated_delivery_date,t.purchasing_date,DAY),0) AS diff_purchasing_estimated,#diff b/w est delivery and purchasing
IFNULL(DATE_DIFF(t.estimated_delivery_date,t.delivery_date,DAY),0) AS diff_estimated_delivery # diff b/w estimated delivery and delivery date
FROM
(SELECT customer_id,order_id, EXTRACT(date FROM order_purchase_timestamp) AS purchasing_date,
EXTRACT(date FROM order_delivered_customer_date) AS delivery_date,
EXTRACT(date FROM order_estimated_delivery_date) AS estimated_delivery_date
FROM 'Target_SQL_Business_Case.orders'
ORDER BY purchasing_date) t
```

#### Top rows in the output

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_id	order_id	diff_purchasing_delivery	diff_purchasing_estimated	diff_estimated_delivery	
1	08c5351a6...	2e7a8482f...	0	46	0	
2	683c54fc24...	e5fa5a721...	0	53	0	
3	622e13439...	809a282bb...	0	17	0	
4	86dc2ffce2...	bfb0f9bd...	55	19	-36	
5	b106b360fe...	71303d7e9...	0	23	0	
6	70fc57eeae...	65d1e226d...	36	53	17	
7	7ec40b225...	be5bc2f0d...	24	35	11	
8	e6f959bf38...	ae8a60e4b...	31	59	28	
9	7812fcebfc...	cd3b8574c...	11	51	40	
10	b8cf418e97...	d207cc272...	28	51	23	
11	6f98933271...	a41c8759f...	31	57	26	
12	dc607dc98...	ef1b29b59...	29	53	24	
13	355077684...	3b697a20d...	23	24	1	
14	2ff5a64555...	5cb8558cb...	0	59	0	
15	68f4ad79cc...	ddaec6fff9...	0	43	0	

Results per page: 50 1 - 50 of 99441

### 5.2 , Find time\_to\_delivery & diff\_estimated\_delivery.

Answer 5.2, We will do the following steps:

Step 1: We will calculate the  $\text{time\_to\_delivery} = \text{order\_purchase\_timestamp} - \text{order\_delivered\_customer\_date}$

Step 2:  $\text{diff\_estimated\_delivery} = \text{order\_estimated\_delivery\_date} - \text{order\_delivered\_customer\_date}$



```

SELECT t.customer_id,t.order_id,
IFNULL(DATE_DIFF(t.delivery_date,t.purchasing_date,DAY),0)AS time_to_delivery,#difference b/w delivery date - purchasing date = time to delivery
IFNULL(DATE_DIFF(t.estimated_delivery_date,t.delivery_date,DAY),0) AS diff_estimated_delivery # diff b/w estimated delivery and delivery date
FROM
(SELECT customer_id,order_id, EXTRACT(date FROM order_purchase_timestamp) AS purchasing_date,
EXTRACT(date FROM order_delivered_customer_date) AS delivery_date,
EXTRACT(date FROM order_estimated_delivery_date) AS estimated_delivery_date
FROM 'Target_SQL_Business_Case.orders'
ORDER BY purchasing_date) t

```

Row	customer_id	order_id	time_to_delivery	diff_estimated_delivery
1	08c5351a6...	2e7a8482f...	0	0
2	683c54fc24...	e5fa5a721...	0	0
3	622e13439...	809a282b...	0	0
4	86dc2ffce2...	bfb0f9bd...	55	-36
5	b106b360fe...	71303d7e...	0	0
6	70fc57eeae...	65d1e226...	36	17
7	7ec40b225...	be5bc2f0d...	24	11
8	e6f959bf38...	ae8a60e4...	31	28
9	7812fcebfc...	cd3b8574...	11	40
10	b8cf418e97...	d207cc27...	28	23

5.3, GROUP data by state, take mean of freight\_value, time\_to\_delivery , diff\_estimated\_delivery.

**Answer 5.3, We will do the following steps:**

**Step 1: We will consider the below tables:**

- a) customers for state
- b) order\_items for freight\_value
- c) orders for time\_to\_delivery, diff\_estimated\_delivery and we can use the query of 5.2 question.

**Step 2: We will perform JOIN on the above tables and we will also use CTEs to solve this query.**

```

with delivery_details as #CTE for the details related to delivery
(SELECT t.customer_id,t.order_id,
IFNULL(DATE_DIFF(t.delivery_date,t.purchasing_date,DAY),0)AS time_to_delivery,#difference b/w delivery date - purchasing date = time to delivery
IFNULL(DATE_DIFF(t.estimated_delivery_date,t.delivery_date,DAY),0) AS diff_estimated_delivery # diff b/w estimated delivery and delivery date
FROM
(SELECT customer_id,order_id, EXTRACT(date FROM order_purchase_timestamp) AS purchasing_date,
EXTRACT(date FROM order_delivered_customer_date) AS delivery_date,
EXTRACT(date FROM order_estimated_delivery_date) AS estimated_delivery_date
FROM 'Target_SQL_Business_Case.orders'
ORDER BY purchasing_date) t),
freight_details as #CTE for freight related details and order id is primary key here
(SELECT order_id,freight_value
FROM 'Target_SQL_Business_Case.order_items')

SELECT c.customer_state,ROUND(AVG(f.freight_value)) AS avg_freight,
ROUND(AVG(d.time_to_delivery)) AS avg_delivery_time,
ROUND(AVG(d.diff_estimated_delivery)) AS avg_estimated_delivery
FROM 'Target_SQL_Business_Case.customers' c JOIN delivery_details d ON c.customer_id = d.customer_id
JOIN freight_details f ON d.order_id = f.order_id
GROUP BY c.customer_state
ORDER BY avg_freight DESC
LIMIT 5

```

5.4 , SORT the data to get the following

5.5 , Top 5 states with highest/lowest freight value.

**Answer 5.5**

✓ The top 5 states with highest freight\_value and refer above query in 5.3 to get this result.

Row	customer_state	avg_freight	avg_delivery_time	avg_estimated_delivery
1	RR	43.0	25.0	16.0
2	PB	43.0	20.0	13.0
3	RO	41.0	19.0	20.0
4	AC	40.0	20.0	21.0
5	PI	39.0	19.0	11.0

✓ The top 5 states with lowest freight\_value, to get the lowest freight\_value, we will ORDER BY in ascending order and LIMIT by 5.

Row	customer_state	avg_freight	avg_delivery_time	avg_estimated_delivery
1	SP	15.0	8.0	11.0
2	PR	21.0	12.0	13.0
3	RJ	21.0	15.0	12.0
4	DF	21.0	13.0	12.0
5	MG	21.0	12.0	13.0

5.6, Top 5 states with highest/lowest average delivery time.

Answer 5.6, In this question we will just ORDER BY average\_delivery\_time.

✓ The following are the states with highest average delivery time, here we just ORDER BY average\_delivery\_time in descending order.

Row	customer_state	avg_freight	avg_delivery_time	avg_estimated_delivery
1	AP	34.0	28.0	18.0
2	AM	33.0	26.0	20.0
3	RR	43.0	25.0	16.0
4	AL	36.0	24.0	8.0
5	PA	36.0	23.0	14.0

✓ The following are the states with lowest average delivery time, here we just ORDER BY average\_delivery\_time in ascending order.

Row	customer_state	avg_freight	avg_delivery_time	avg_estimated_delivery
1	SP	15.0	8.0	11.0
2	MG	21.0	12.0	13.0
3	PR	21.0	12.0	13.0
4	DF	21.0	13.0	12.0
5	RS	22.0	15.0	14.0

5.7, Top 5 states where delivery is really fast/not so fast compared to estimated date.

Answer 5.7, We will do the following steps:

Step 1: Finding fast delivery = average\_delivery\_time < avg\_estimated\_delivery, and we will get the following results where there's fast delivery and to get top 5 we will ORDER BY average\_delivery\_time in ascending order.

NOTE: We will use the WHERE clause for doing the above comparison

```

SELECT table_1.customer_state,table_1.avg_delivery_time,table_1.avg_estimated_delivery #Calculating the fast delivery
FROM
  (SELECT c.customer_state,ROUND(AVG(f.freight_value)) AS avg_freight,
    ROUND(AVG(d.time_to_delivery)) as avg_delivery_time,
    ROUND(AVG(d.diff_estimated_delivery) )AS avg_estimated_delivery
  FROM `Target_SQL_Business_Case.customers` c JOIN delivery_details d ON c.customer_id = d.customer_id
  JOIN freight_details f ON d.order_id = f.order_id
  GROUP BY c.customer_state) table_1
WHERE table_1.avg_delivery_time < table_1.avg_estimated_delivery
ORDER BY table_1.avg_delivery_time

```

Row	customer_state	avg_delivery_time	avg_estimated_delivery
1	SP	8.0	11.0
2	MG	12.0	13.0
3	PR	12.0	13.0
4	RO	19.0	20.0
5	AC	20.0	21.0

#### OBSERVATION :

There are only five states where average delivery is faster than average estimated delivery and hence we don't need to use LIMIT BY 5 clause.

Step 2: Finding not so fast delivery = average\_delivery\_time > avg\_estimated\_delivery, and we will get the following results where there's fast delivery and to get top 5 we will ORDER BY average\_delivery\_time in descending order.

NOTE: We will use the WHERE clause for doing the above comparison

```

SELECT table_1.customer_state,table_1.avg_delivery_time,table_1.avg_estimated_delivery #Calculating the not so fast delivery
FROM
  (SELECT c.customer_state,ROUND(AVG(f.freight_value)) AS avg_freight,
    ROUND(AVG(d.time_to_delivery)) as avg_delivery_time,
    ROUND(AVG(d.diff_estimated_delivery) )AS avg_estimated_delivery
  FROM `Target_SQL_Business_Case.customers` c JOIN delivery_details d ON c.customer_id = d.customer_id
  JOIN freight_details f ON d.order_id = f.order_id
  GROUP BY c.customer_state) table_1
WHERE table_1.avg_delivery_time > table_1.avg_estimated_delivery
ORDER BY table_1.avg_delivery_time DESC #Here we ordering by avg delivery time in descending order
LIMIT 5 #There are 22 such states that are not so fast so we selecting top 5 not so fast states

```

Row	customer_state	avg_delivery_time	avg_estimated_delivery
1	AP	28.0	18.0
2	AM	26.0	20.0
3	RR	25.0	16.0
4	AL	24.0	8.0
5	PA	23.0	14.0

## 6. Payment type analysis:

### 6.1, Month over Month count of orders for different payment types.

Answer 6.1, We will do the following steps:

Step 1: We will consider the following tables:

- orders for months over months details.
- payments for payment\_type details.

Step 2: We will count orders based on months first and then we will count orders based on payment type.

```
SELECT *
FROM
  (SELECT p.payment_type, EXTRACT(month FROM o.order_purchase_timestamp) as months, COUNT(o.order_id) as total_order_by_month
   FROM 'Target_SQL_Business_Case.orders' o JOIN 'Target_SQL_Business_Case.payments' p ON o.order_id = p.order_id
   GROUP BY p.payment_type, EXTRACT(month FROM o.order_purchase_timestamp)) t1
ORDER BY t1.payment_type, t1.months
```

Row	payment_type	month	total_order_by_month
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6103
14	credit_card	2	6609
15	credit_card	3	7707
16	credit_card	4	7301
17	credit_card	5	6350
18	credit_card	6	7276
19	credit_card	7	7841
20	credit_card	8	8289



Paym..		Months											
credit_card	Total Order	10K	6103	6609	7707	7301	8350	7276	7841	8269			
	5K										3286	3778	5897
debit_card	Total Order	10K											
	5K		118	82	109	124	81	209	264	311	43	54	70
not_defined	Total Order	10K											
	5K									2	1		
UPI	Total Order	10K											
	5K		1715	1723	1942	1783	2035	1807	2074	2077	903	1056	1509
voucher	Total Order	10K											
	5K		477	424	591	572	613	563	645	589	302	318	387
			January	February	March	April	May	June	July	August	Septemb..	October	November
													December

## OBSERVATION:

We can see that majority of the customers have credit card as the mode of payment in Brazil.

## 6.2, Count of orders based on the number of payment installments.

Answer 6.2, We will do the following steps:

Step 1: We will select payment installments data from the payments table

Step 2: We will use COUNT on order\_id using GROUP BY.

```
SELECT payment_installments, COUNT(order_id) as Total_customers
FROM `Target_SQL_Business_Case.payments`
GROUP BY payment_installments
```

Row	payment_installments	Total_customers
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
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15
16	15	74
17	16	5
18	17	8
19	18	27
20	20	17
21	21	3
22	22	1
23	23	1
24	24	18

## 7. Actionable insights:

Answer 7, The following are that insights that we got from our analysis.

▶ **Performance of the company:** We have two year data of the company and we can see that during the initial years there we small number of customers but tremendously got increased in the later months. Hence the market capitalization of the company is robust and increasing.

▶ **Trend of customers:** The customers are increasing as we can see that and in month of November 2017, company had the maximum number of customers and overall we have increasing trend of customers which shows company's good health.

▶ **Peak time in which customers buy:** We also observed that in the evening or night time the Brazilian customers tend to buy a lot as compared to other time periods. This is because most people naturally tend to buy products when they're free.

▶ **State with highest number of customers:** "SP" or São Paulo have the maximum number of customers in Brazil and it makes sense because it is largest populated state in Brazil hence it'll have largest customers.

▶ **Average total revenue of customers by state:** If you think that state with maximum customers will give us more average revenue then you're wrong. While performing our analysis we found that the state "RR" or Roraima has the highest average revenue(price+freight\_value).

▶ **Delivery time and freight relationship:** The orders or states which have more delivery time have more freight\_value and vice versa because logically if delivery taking more time then seller and customers live far away and seller will incur transportation cost or freight\_value which will be high.

▶ **Payment analysis:** We can see that most of the customers tend to do shopping via credit card and the installments they take is 0,1,2 or 3. There are few number of customers who are taking long term installments.

▶ **Delivery analysis:** There are only 5 states where average\_delivery\_time is less than average estimated delivery time and remaining 22 states have average\_delivery\_time greater than average estimated delivery time.

---

## 8. Recommendations:

Answer 8, The following recommendations will help the company to grow faster:

🚩 **Discounts and offers:** As we observe that most of the customers tend to buy at evening or night therefore that time period is our rush hour and we should give more offers and discounts during night. The last quarter of year also have high number of customers and company should provide festival kind of sales during that period.

🚩 **Increasing reachability:** The company should provide more facilities where average revenue is high cause these are potential hotspots for earning more money and customers.

🚩 **Figuring out loses:** The "SP" state have low average revenue but it has maximum number of customers, company should figure out the reason for this low average revenue and eliminate the factors causing low average revenue.

🚩 **Better delivery facilities:** Company should try to reduce the time of delivery. Currently there are only 5 states where average\_delivery\_time is less than estimated delivery time. Company should work in those 22 states to reduce delivery time.

🚩 **More credit card benefits:** As we observed that most of the customers loves credit card purchase so company should include more credit card benefits.

🚩 **Subscription plan:** Company should provide subscription plans to regular customers where there's could get exclusive offers.

🚩 **Including more sellers on platform:** Company can include more number of trustworthy sellers which could provide diverse range of products to customers.

---



