Business Case: Target SQL

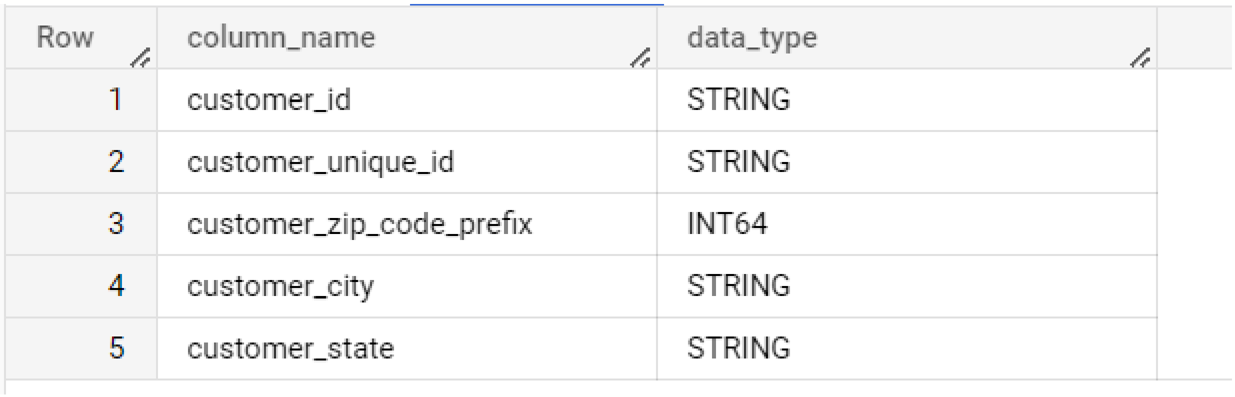
Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

Q1: Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1. Data type of columns in a table

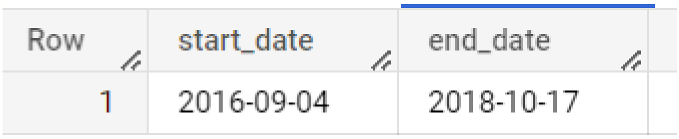
```
SELECT column_name, data_type
 FROM `scaler-dsml-sql-384102.Target_SQL.`.INFORMATION_SCHEMA.COLUMNS
 WHERE table_name="customers";
```



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. Time period for which the data is given

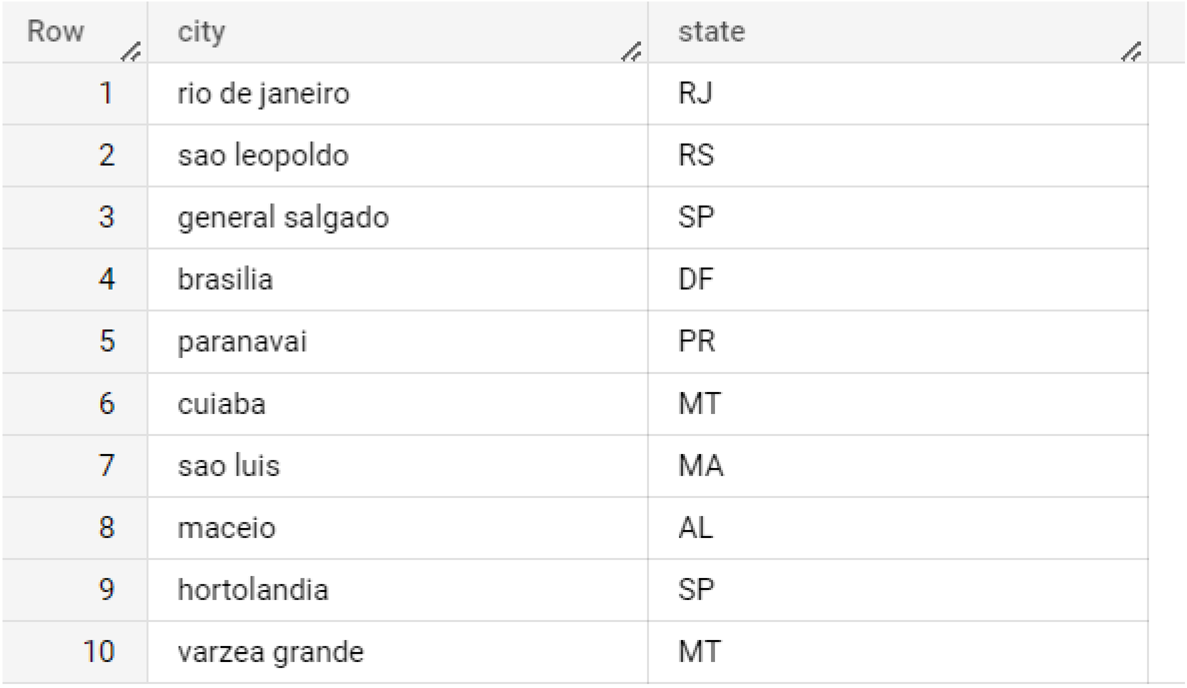
```
SELECT MIN(EXTRACT(DATE FROM order_purchase_timestamp)) AS start_date, MAX(EXTRACT(DA
TE FROM order_purchase_timestamp)) AS end_date
FROM `Target_SQL.orders`
```



Row	start_date	end_date
1	2016-09-04	2018-10-17

3. Cities and States of customers ordered during the given period

```
SELECT DISTINCT customer_city AS city, customer_state AS state
FROM `Target_SQL.customers` AS c RIGHT JOIN `Target_SQL.orders` AS o ON c.customer_id
 = o.customer_id
WHERE order_purchase_timestamp BETWEEN "2016-09-04" AND "2018-10-17"
```

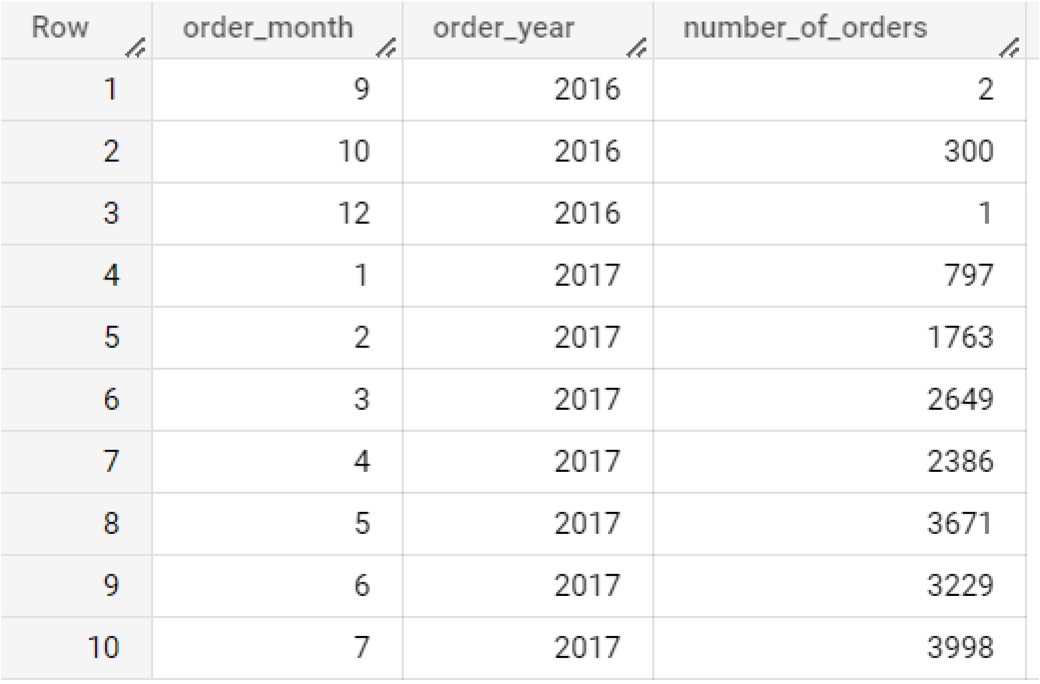


Row	city	state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT

Q2. In-depth Exploration

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?

```
SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
       EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       COUNT(order_id) AS number_of_orders
FROM `Target_SQL.orders`
WHERE order_status!="canceled"
GROUP BY order_month, order_year
ORDER BY order_year, order_month;
```

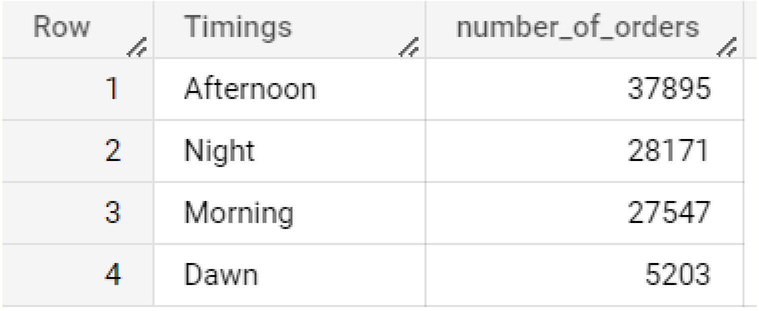


Row	order_month	order_year	number_of_orders
1	9	2016	2
2	10	2016	300
3	12	2016	1
4	1	2017	797
5	2	2017	1763
6	3	2017	2649
7	4	2017	2386
8	5	2017	3671
9	6	2017	3229
10	7	2017	3998

- Yes there is a growing trend in the e-commerce business in Brazil and is steadily growing year by year.
- Seasonality can be seen as the sales tend to increase before and after the Christmas holidays i.e during November and January

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

```
SELECT (CASE WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN "D
awn"
             WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN "M
orning"
             WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN "
Afternoon"
             WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN "
Night"
             END) AS Timings,
       COUNT(order_id) AS number_of_orders
FROM `Target_SQL.orders`
WHERE order_status!="canceled"
GROUP BY timings
ORDER BY number_of_orders DESC;
```



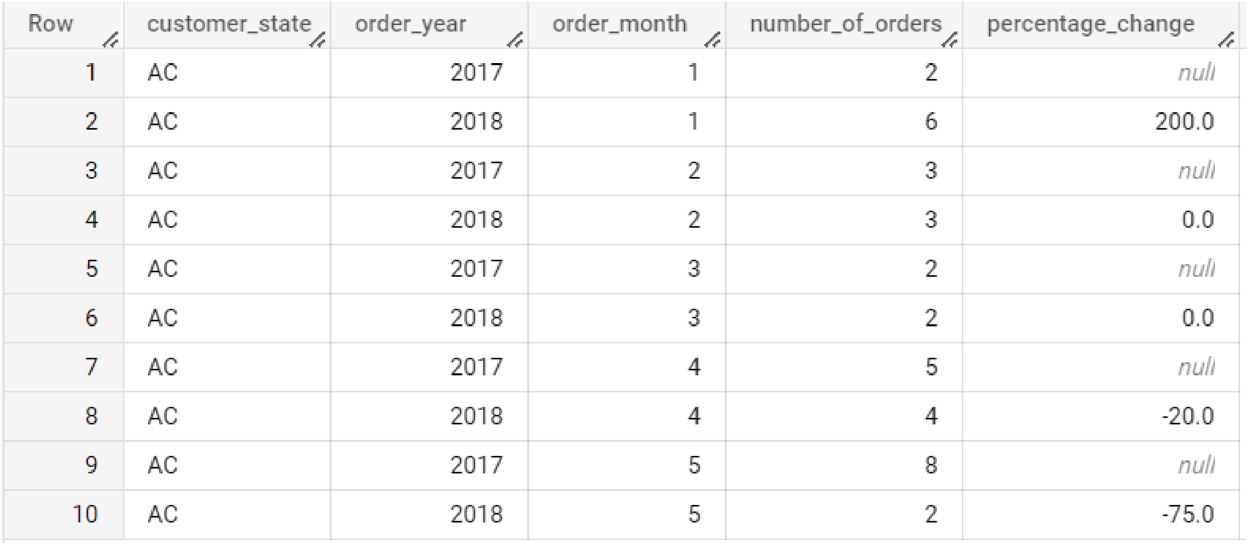
Row	Timings	number_of_orders
1	Afternoon	37895
2	Night	28171
3	Morning	27547
4	Dawn	5203

- Most of the Brazilian customers like to buy during the Afternoon than in Dawn. And also they like to shop during Night and Mornings as well.

Q3. Evolution of E-commerce orders in the Brazil region

1. Get month on month orders by states

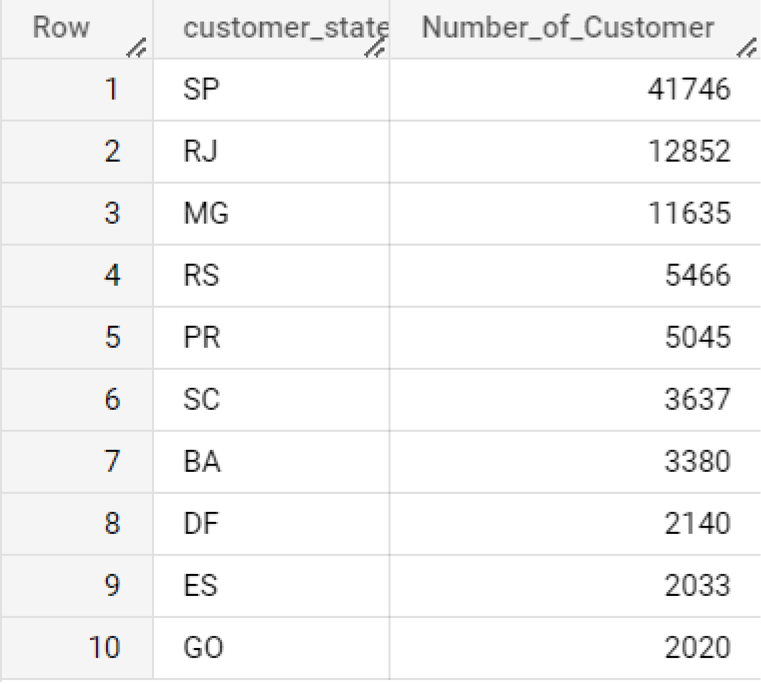
```
    with order_data AS
(
SELECT *,
 EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
 EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year
 FROM `Target_SQL.orders`
 )

SELECT
 customer_state,
 order_year,
 order_month,
 COUNT(o.order_id) AS number_of_orders,
 ROUND((COUNT(o.order_id)-
LAG(COUNT(o.order_id)) OVER (PARTITION BY customer_state, order_month ORDER BY order_month,order_year))/LAG(COUNT(o.order_id)) OVER (PARTITION BY customer_state ORDER BY
order_month,order_year)*100,2) AS percentage_change
 FROM order_data AS o LEFT JOIN `Target_SQL.customers` AS c ON o.customer_id=c.customer_id
 WHERE order_status = "delivered"
 GROUP BY customer_state, order_year, order_month
 ORDER BY customer_state, order_month, order_year
```



Row	customer_state	order_year	order_month	number_of_orders	percentage_change
1	AC	2017	1	2	null
2	AC	2018	1	6	200.0
3	AC	2017	2	3	null
4	AC	2018	2	3	0.0
5	AC	2017	3	2	null
6	AC	2018	3	2	0.0
7	AC	2017	4	5	null
8	AC	2018	4	4	-20.0
9	AC	2017	5	8	null
10	AC	2018	5	2	-75.0

2. Distribution of customers across the states in Brazil

```
SELECT customer_state,
       COUNT(customer_id) AS Number_of_Customer
FROM `Target_SQL.customers`
GROUP BY customer_state
ORDER BY Number_of_Customer DESC
```

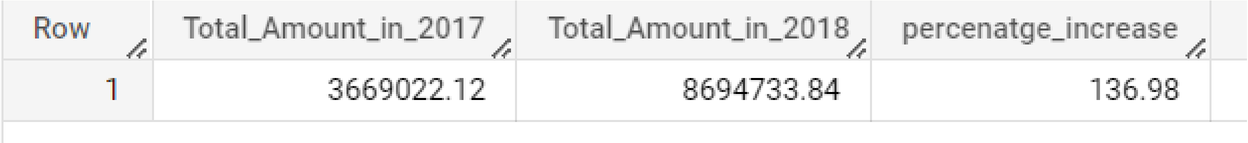


Row	customer_state	Number_of_Customer
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table

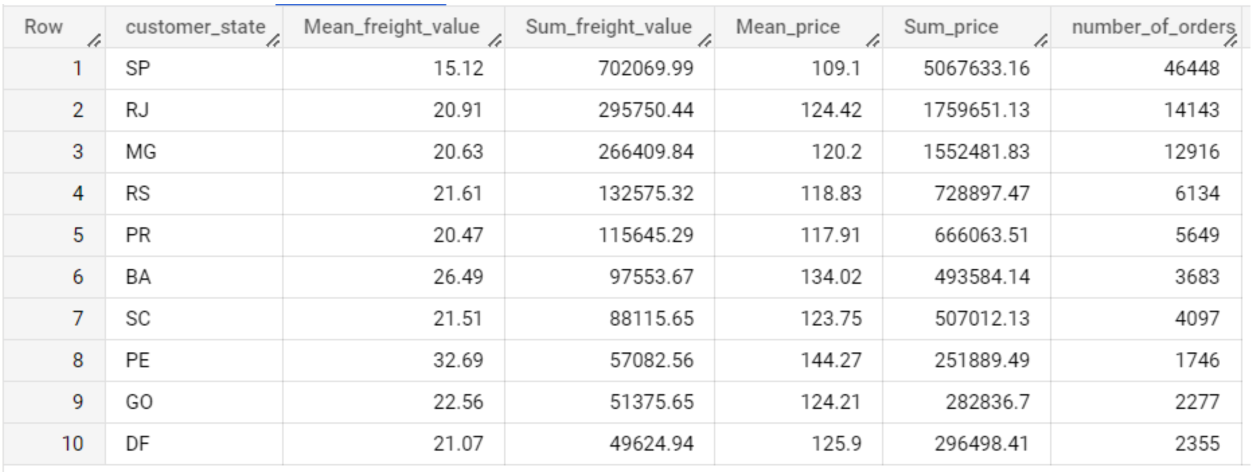
```
WITH orders_monthly AS (
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
ROUND(SUM(payment_value),2) AS Total_Amount
FROM `Target_SQL.orders` AS o LEFT JOIN `Target_SQL.payments` AS p ON o.order_id=p.or
der_id
WHERE EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8 AND
EXTRACT(YEAR FROM order_purchase_timestamp) IN (2017, 2018)
GROUP BY order_year, order_month
ORDER BY order_year, order_month
)

SELECT
SUM(CASE WHEN order_year= 2017 THEN Total_Amount END) AS Total_Amount_in_2017,
SUM(CASE WHEN order_year= 2018 THEN Total_Amount END) AS Total_Amount_in_2018,
ROUND((SUM(CASE WHEN order_year= 2018 THEN Total_Amount END)-
SUM(CASE WHEN order_year= 2017 THEN Total_Amount END))/SUM(CASE WHEN order_year= 2017
 THEN Total_Amount END)*100,2) AS percenatge_increase
FROM orders_monthly
```



Row	Total_Amount_in_2017	Total_Amount_in_2018	percenatge_increase
1	3669022.12	8694733.84	136.98

2. Mean & Sum of price and freight value by customer state

```
SELECT customer_state,
       ROUND(AVG(freight_value),2) AS Mean_freight_value,
       ROUND(SUM(freight_value),2) AS Sum_freight_value,
       ROUND(AVG(price),2) AS Mean_price,
       ROUND(SUM(price),2) AS Sum_price,
       COUNT(*) AS number_of_orders
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  c.customer_id=o.customer_id
WHERE order_status = "delivered"
GROUP BY customer_state
ORDER BY Sum_freight_value DESC, Mean_freight_value DESC
```



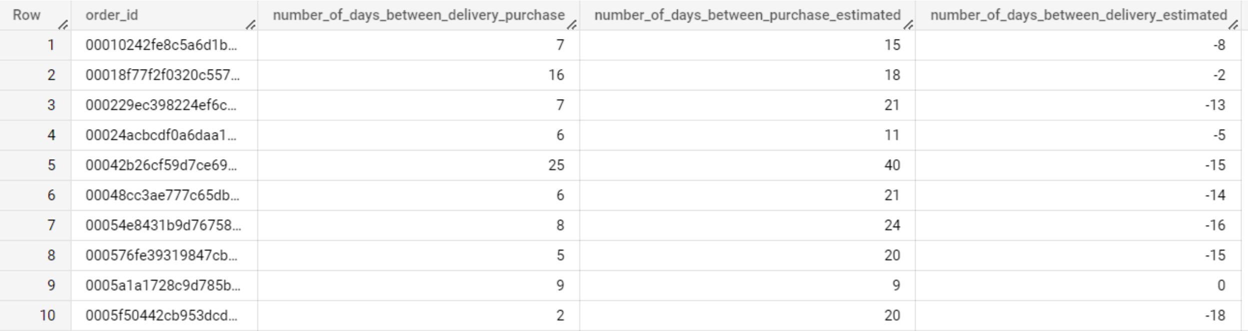
Row	customer_state	Mean_freight_value	Sum_freight_value	Mean_price	Sum_price	number_of_orders
1	SP	15.12	702069.99	109.1	5067633.16	46448
2	RJ	20.91	295750.44	124.42	1759651.13	14143
3	MG	20.63	266409.84	120.2	1552481.83	12916
4	RS	21.61	132575.32	118.83	728897.47	6134
5	PR	20.47	115645.29	117.91	666063.51	5649
6	BA	26.49	97553.67	134.02	493584.14	3683
7	SC	21.51	88115.65	123.75	507012.13	4097
8	PE	32.69	57082.56	144.27	251889.49	1746
9	GO	22.56	51375.65	124.21	282836.7	2277
10	DF	21.07	49624.94	125.9	296498.41	2355

- From the output we can get to the conclusion that as the average freight cost for a state increases the number of orders decreases.
- SP state has the least average freight cost with highest number of orders and state RR has highest average freight cost.

Q 5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

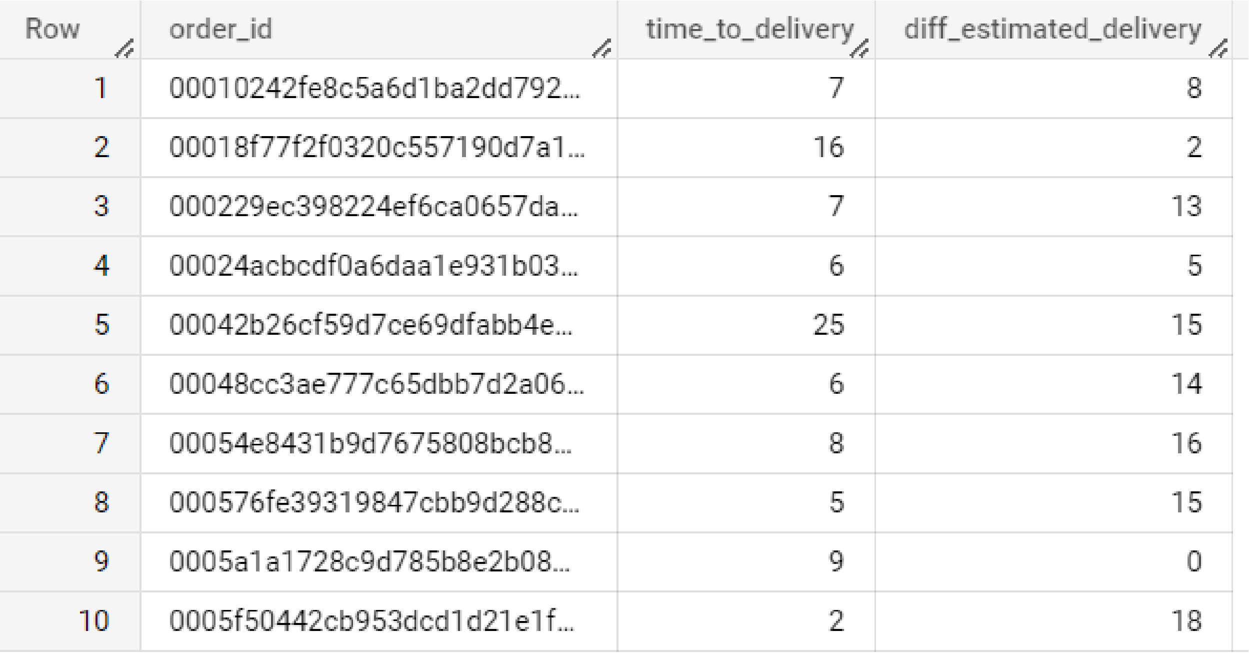
```sql
SELECT
order_id,
DATETIME_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS number_of_days_between_delivery_purchase,
DATETIME_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS number_of_days_between_purchase_estimated,
DATETIME_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) AS number_of_days_between_delivery_estimated
FROM `Target_SQL.orders`
WHERE order_status="delivered"
ORDER BY order_id;
```



Row	order_id	number_of_days_between_delivery_purchase	number_of_days_between_purchase_estimated	number_of_days_between_delivery_estimated
1	00010242fe8c5a6d1b...	7	15	-8
2	00018f77f2f0320c557...	16	18	-2
3	000229ec398224ef6c...	7	21	-13
4	00024acbcdf0a6daa1...	6	11	-5
5	00042b26cf59d7ce69...	25	40	-15
6	00048cc3ae777c65db...	6	21	-14
7	00054e8431b9d76758...	8	24	-16
8	000576fe39319847cb...	5	20	-15
9	0005a1a1728c9d785b...	9	9	0
10	0005f50442cb953dcd...	2	20	-18

2. Find time_to_delivery & diff_estimated_delivery.

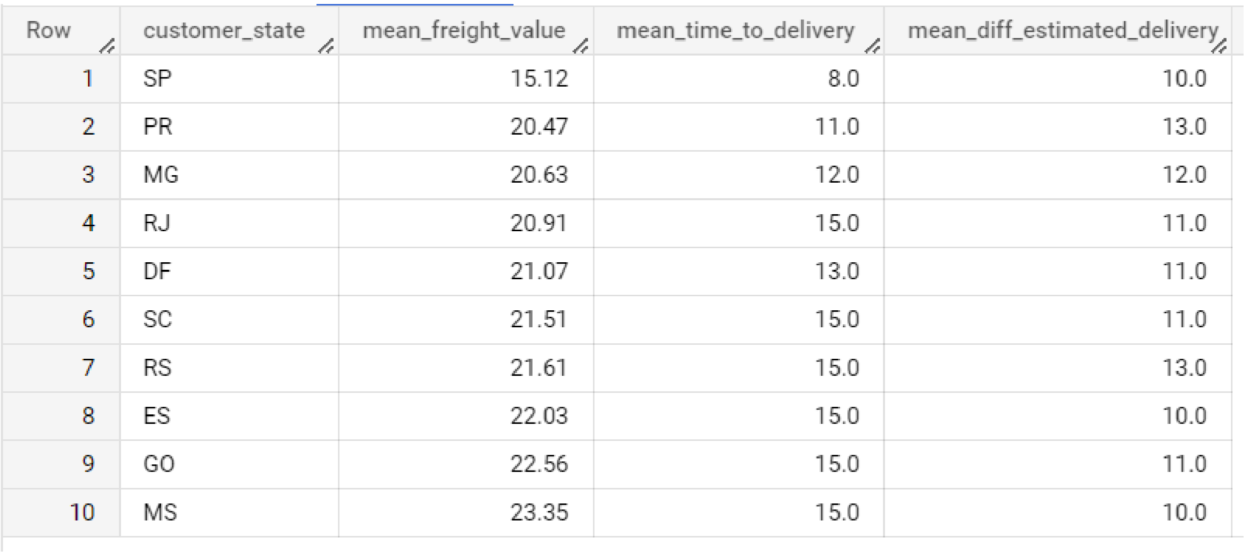
```
SELECT
order_id,
DATETIME_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_t
o_delivery,
DATETIME_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS d
iff_estimated_delivery
FROM `Target_SQL.orders`
WHERE order_status="delivered"
ORDER BY order_id;
```



Row	order_id	time_to_delivery	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
WITH order_data AS
(
SELECT
order_id, customer_id,
DATETIME_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_to_delivery,
DATETIME_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estimated_delivery
FROM `Target_SQL.orders`
WHERE order_status="delivered"
ORDER BY order_id
)

SELECT customer_state,
       ROUND(AVG(freight_value),2) AS mean_freight_value,
       ROUND(AVG(time_to_delivery)) AS mean_time_to_delivery,
       ROUND(AVG(diff_estimated_delivery)) AS mean_diff_estimated_delivery
FROM order_data AS o LEFT JOIN `Target_SQL.customers` AS c ON o.customer_id=c.customer_id
LEFT JOIN `Target_SQL.order_items` AS p ON o.order_id=p.order_id
GROUP BY customer_state
ORDER BY mean_freight_value
```

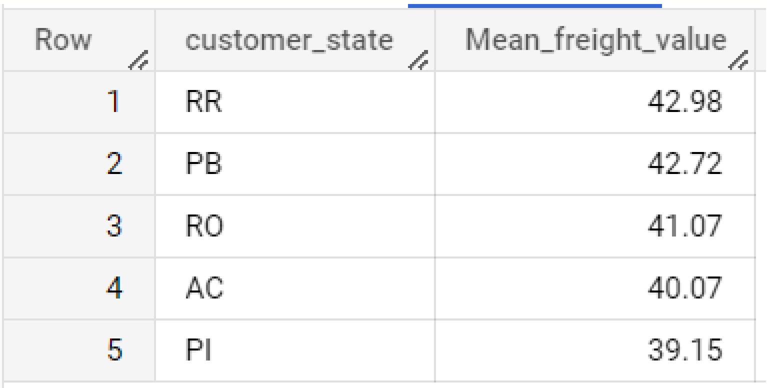


Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.12	8.0	10.0
2	PR	20.47	11.0	13.0
3	MG	20.63	12.0	12.0
4	RJ	20.91	15.0	11.0
5	DF	21.07	13.0	11.0
6	SC	21.51	15.0	11.0
7	RS	21.61	15.0	13.0
8	ES	22.03	15.0	10.0
9	GO	22.56	15.0	11.0
10	MS	23.35	15.0	10.0

4. Sort the data to get the following:

5.a) Top 5 states with highest average freight value:

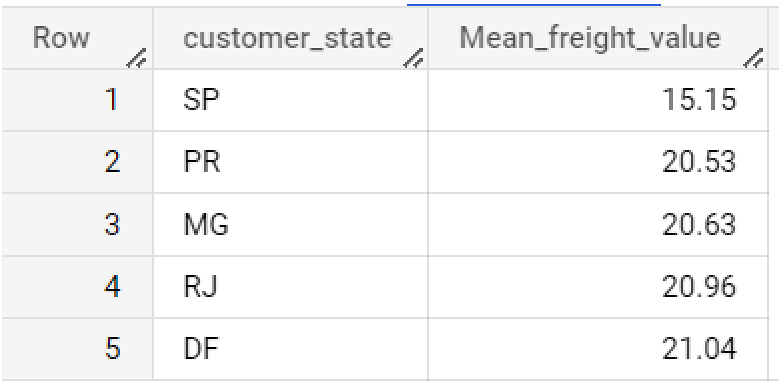
```
SELECT customer_state, ROUND(AVG(freight_value),2) AS Mean_freight_value
FROM `Target_SQL.orders` AS o INNER JOIN `Target_SQL.order_items` AS oi ON o.order_id
=oi.order_id
INNER JOIN `Target_SQL.customers` AS c ON  c.customer_id=o.customer_id
GROUP BY customer_state
ORDER BY Mean_freight_value DESC
LIMIT 5
```



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

5.b) Top 5 states with lowest average freight value:

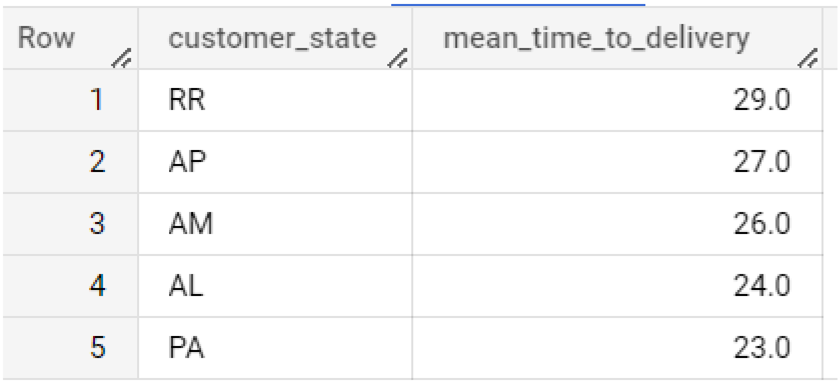
```
SELECT customer_state, ROUND(AVG(freight_value),2) AS Mean_freight_value
FROM `Target_SQL.orders` AS o INNER JOIN `Target_SQL.order_items` AS oi ON o.order_id
=oi.order_id
INNER JOIN `Target_SQL.customers` AS c ON  c.customer_id=o.customer_id
GROUP BY customer_state
ORDER BY Mean_freight_value
LIMIT 5
```



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

6.a) Top 5 states with highest average time to delivery

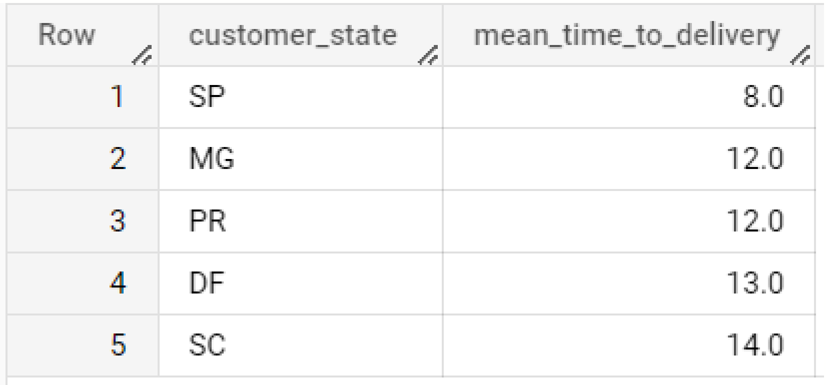
```
SELECT customer_state,
       ROUND(AVG(DATETIME_DIFF(order_delivered_customer_date, order_purchase_timestam
p, DAY))) AS mean_time_to_delivery,
FROM `Target_SQL.orders` AS o LEFT JOIN `Target_SQL.customers` AS c ON o.customer_id=
c.customer_id
GROUP BY customer_state
ORDER BY mean_time_to_delivery DESC
LIMIT 5
```



Row	customer_state	mean_time_to_delivery
1	RR	29.0
2	AP	27.0
3	AM	26.0
4	AL	24.0
5	PA	23.0

6.b) Top 5 states with lowest average time to delivery

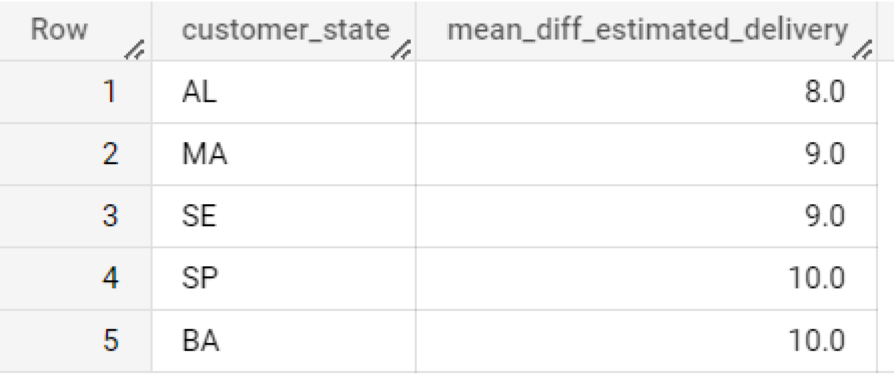
```
SELECT customer_state,
       ROUND(AVG(DATETIME_DIFF(order_delivered_customer_date, order_purchase_timestam
p, DAY))) AS mean_time_to_delivery,
FROM `Target_SQL.orders` AS o LEFT JOIN `Target_SQL.customers` AS c ON o.customer_id=
c.customer_id
GROUP BY customer_state
ORDER BY mean_time_to_delivery
LIMIT 5
```



Row	customer_state	mean_time_to_delivery
1	SP	8.0
2	MG	12.0
3	PR	12.0
4	DF	13.0
5	SC	14.0

7.a) Top 5 states where delivery is really fast compared to estimated date

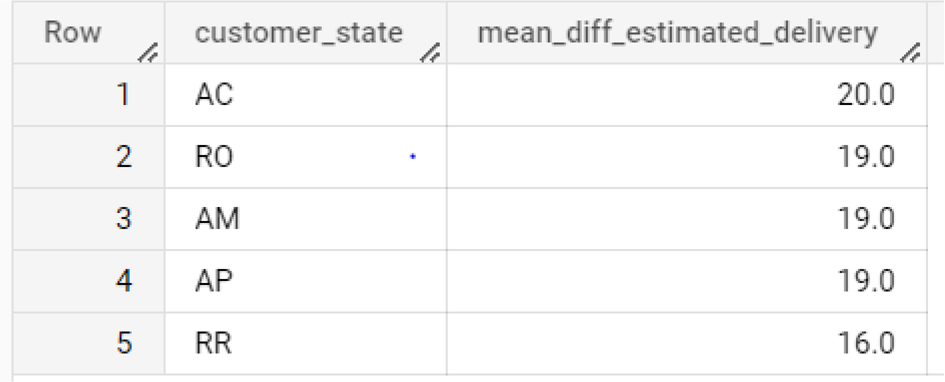
```
SELECT customer_state,
       ROUND(AVG(DATETIME_DIFF(order_estimated_delivery_date, order_delivered_custome
r_date, DAY))) AS diff_estimated_delivery
FROM `Target_SQL.orders` AS o LEFT JOIN `Target_SQL.customers` AS c ON o.customer_id=
c.customer_id
GROUP BY customer_state
ORDER BY diff_estimated_delivery
LIMIT 5
```



Row	customer_state	mean_diff_estimated_delivery
1	AL	8.0
2	MA	9.0
3	SE	9.0
4	SP	10.0
5	BA	10.0

7.a) Top 5 states where delivery is not so fast compared to estimated date

```
SELECT customer_state,
       ROUND(AVG(DATETIME_DIFF(order_estimated_delivery_date, order_delivered_custome
r_date, DAY))) AS mean_diff_estimated_delivery
FROM `Target_SQL.orders` AS o LEFT JOIN `Target_SQL.customers` AS c ON o.customer_id=
c.customer_id
GROUP BY customer_state
ORDER BY mean_diff_estimated_delivery DESC
LIMIT 5
```

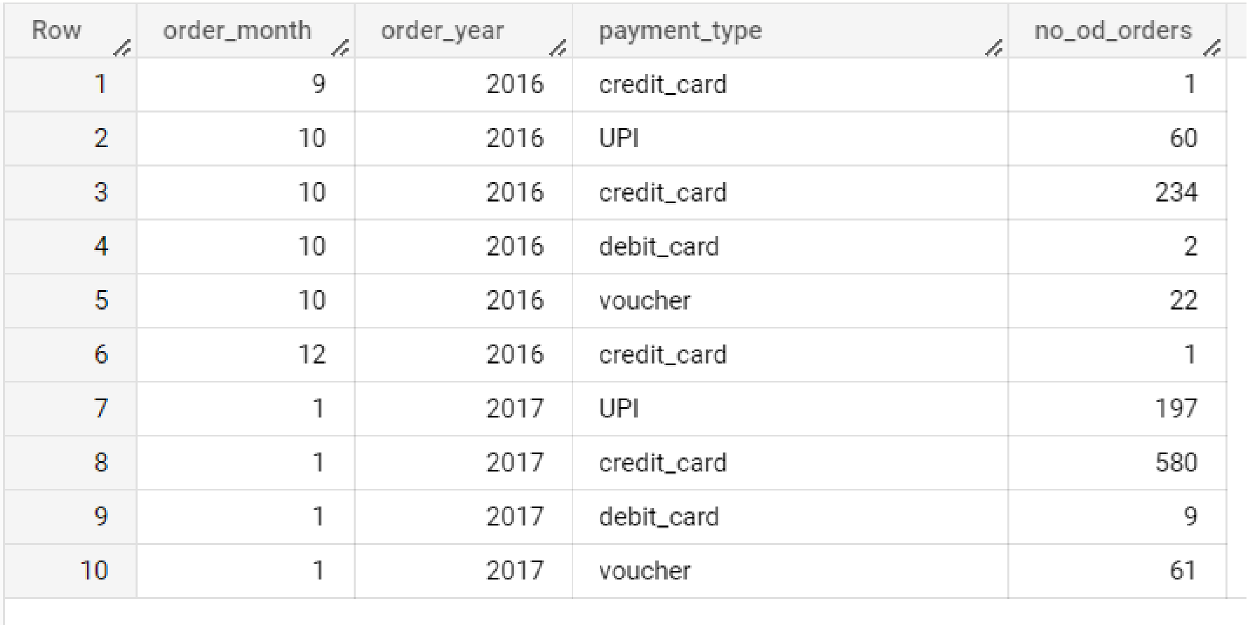


Row	customer_state	mean_diff_estimated_delivery
1	AC	20.0
2	RO	19.0
3	AM	19.0
4	AP	19.0
5	RR	16.0

Q6. Payment type analysis

1. Month over Month count of orders for different payment types

```
SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
       EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       payment_type,
       COUNT(p.order_id) AS no_od_orders
FROM `Target_SQL.orders` AS o RIGHT JOIN `Target_SQL.payments` AS p ON o.order_id=p.o
rder_id
WHERE order_status!="canceled"
GROUP BY payment_type, order_month, order_year
ORDER BY order_year, order_month, payment_type
```

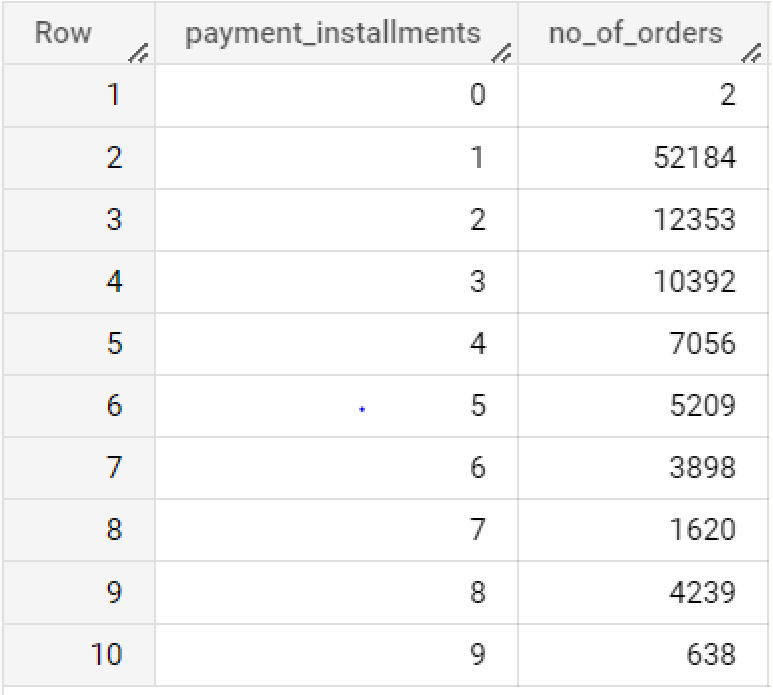


Row	order_month	order_year	payment_type	no_od_orders
1	9	2016	credit_card	1
2	10	2016	UPI	60
3	10	2016	credit_card	234
4	10	2016	debit_card	2
5	10	2016	voucher	22
6	12	2016	credit_card	1
7	1	2017	UPI	197
8	1	2017	credit_card	580
9	1	2017	debit_card	9
10	1	2017	voucher	61

- It can be conclude that most of the Brazilian customers choose credit card for their payments.

2. Count of orders based on the no. of payment installments

```
SELECT payment_installments,
       COUNT(p.order_id) AS no_of_orders
FROM `Target_SQL.orders` AS o RIGHT JOIN `Target_SQL.payments` AS p ON o.order_id=p.o
rder_id
WHERE order_status!="canceled"
GROUP BY payment_installments
ORDER BY payment_installments
```



Row	payment_installments	no_of_orders
1	0	2
2	1	52184
3	2	12353
4	3	10392
5	4	7056
6	5	5209
7	6	3898
8	7	1620
9	8	4239
10	9	638

Recommendations:

- As the Brazilian customers like to shop during the afternoon, the company should provide more offers and discounts during afternoon so that number of orders can be increased.
- The company should collaborate with carrier companies that can help to decrease the freight charges and also the time taken for delivery in the states with high freight charges which can boost the orders from those states.
- As most of the customers use credit cards for payments more offers and discounts should be offered on payments through credit cards so that more orders can be attracted.
- The company should try to create shopping events during the months November, December and January with exciting offers and discounts as most the customers tend to shop more during those months.