Target - Business Case Study (SQL)

Q.1 Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

a) Data type of columns in a table

Table- Customers	
Features	Data type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

Table- Geolocation	
Features	Data type
geolocation_zip_code_prefix	INTEGER
geolocation_lat	FLOAT
geolocation_lng	FLOAT
geolocation_city	STRING
geolocation_state	STRING

Table- order_items	
Features	Data type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT

Table- order_reviews	
Features	Data type
review_id	STRING
order_id	STRING
review_score	INTERGER
review_comment_title	STRING
review_creation_date	TIMESTAMP
review_answer_timestamp	TIMESTAMP

Table- orders	
Features	Data type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

Table- product	
Features	Data type
procuct_id	STRING
product_category	STRING
product_name_length	INTEGER
product_discription_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER

Table- Payments	
Features	Data type
order_id	STRING
payment_sequential	INTEGER
payment_type	STRING
payment_installments	INTEGER
payment_value	FLOAT

Table- seller	
Features	Data type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

Q1 b) Time period for which the data is given

Query: -
```
select min(order_purchase_timestamp) as Min_time,
    max(order_purchase_timestamp) as Max_time
```

```
from Target_analysis.orders
```

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	Min_time	Max_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Q 1 c) Cities and States of customers ordered during the given period.

Query:-

```
select
    distinct c.customer_city,
        c.customer_state
            from Target_analysis.orders as o
            join Target_analysis.Customers as c on o.customer_id=c.customer_id
```

Query results

SAVE RESULTS ▾ EXPL

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_city	customer_state
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC
6	itu	SP
7	jau	SP
8	luz	MG
9	poa	SP
10	uba	MG

Q 2. In-depth Exploration:

a) 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?

Query:-

```
select

    sum(oi.price),
    extract(month from o.order_purchase_timestamp) as Month,
    extract(year from o.order_purchase_timestamp) as Year
from Target_analysis.orders as o
```

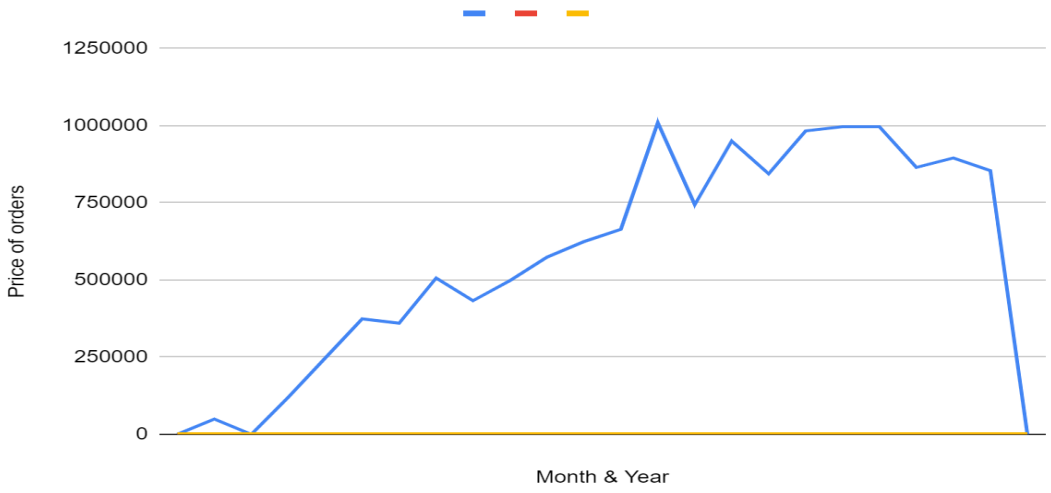
```
left join Target_analysis.order_items as oi
on o.order_id=oi.order_id
group by year,month
order by Year, Month
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	f0_	Month	Year
1	267.36	9	2016
2	49507.6600...	10	2016
3	10.9	12	2016
4	120312.869...	1	2017
5	247303.019...	2	2017
6	374344.300...	3	2017
7	359927.230...	4	2017
8	506071.140...	5	2017
9	433038.600...	6	2017
10	498031.480...	7	2017



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

Query:-

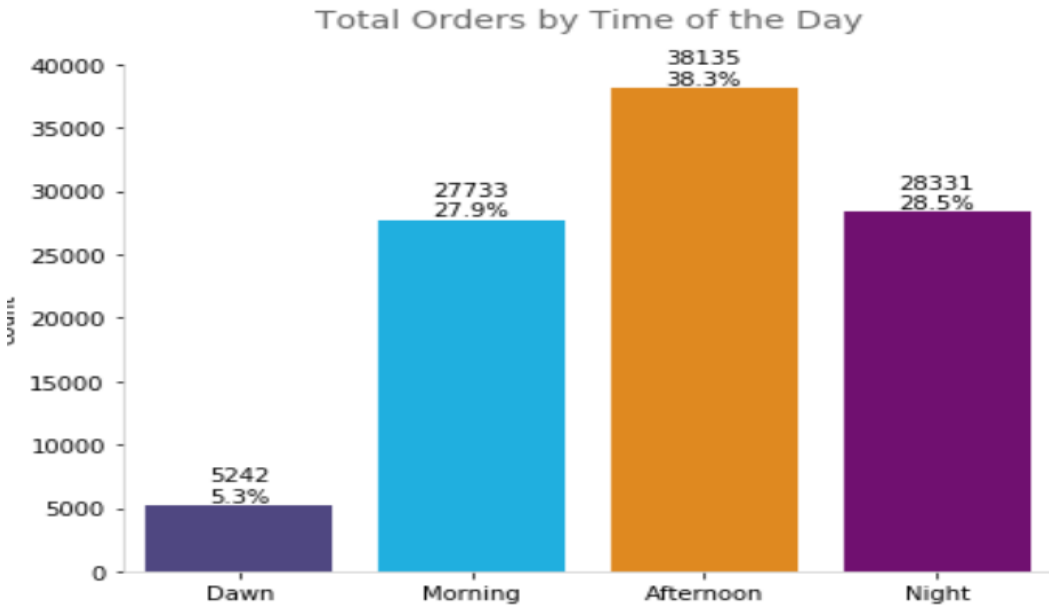
```
select
  count(order_id) as total_order,
  case
  when extract(hour from o.order_purchase_timestamp) between 0 and 6
    then 'Dawn'
  when extract(hour from o.order_purchase_timestamp) between 6 and 12
    then 'Morning'
  when extract(hour from o.order_purchase_timestamp) between 12 and 18
    then 'Afternoon'
  else 'Night'
  end as Time_braket,
from Target_analysis.orders as o
group by Time_braket
order by count(order_id);
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	total_order	Time_braket
1	5242	Dawn
2	27733	Morning
3	28331	Night
4	38135	Afternoon



Q 3) Evolution of E-commerce orders in the Brazil region:

a) Get month on month orders by states

Query:-

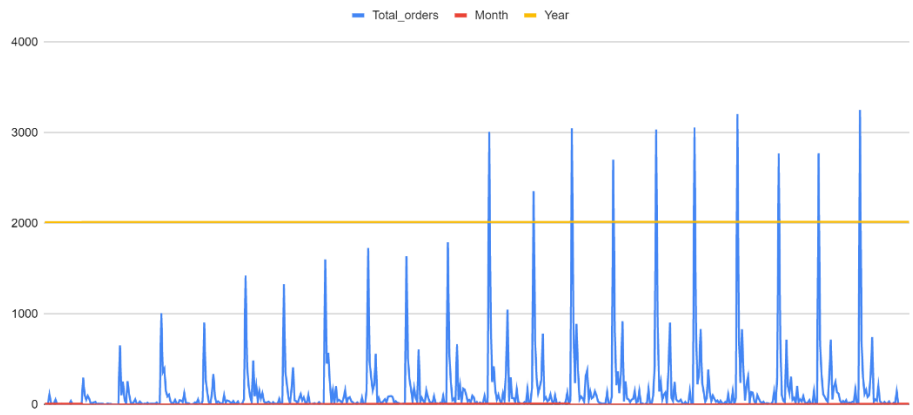
```
select
        count(o.order_id) as Total_orders,
          c.customer_state,
avg(extract(month from o.order_purchase_timestamp)) as Month,
avg(extract(year from o.order_purchase_timestamp)) as Year,
from Target_analysis.Customers as c
left join Target_analysis.orders as o
on c.customer_id=o.customer_id
group by extract(year from o.order_purchase_timestamp), extract(month from o.order_purchase_timesta mp),c.customer_state
order by Year,Month;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	Total_orders	customer_state	Month	Year
1	1	RR	9.0	2016.0
2	1	RS	9.0	2016.0
3	2	SP	9.0	2016.0
4	113	SP	10.0	2016.0
5	24	RS	10.0	2016.0
6	56	RJ	10.0	2016.0
7	3	MT	10.0	2016.0
8	9	GO	10.0	2016.0
9	40	MG	10.0	2016.0
10	8	CE	10.0	2016.0



Q 3 b) Distribution of customers across the states in Brazil.

Query: -

```
select
    customer_state,
    count(customer_id) as total_customers
from Target_analysis.Customers
group by customer_state
order by total_customers
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	total_customers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

Results per page: 50 | 1 – 27 of 27

Q 4 Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

a) 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

Query :-

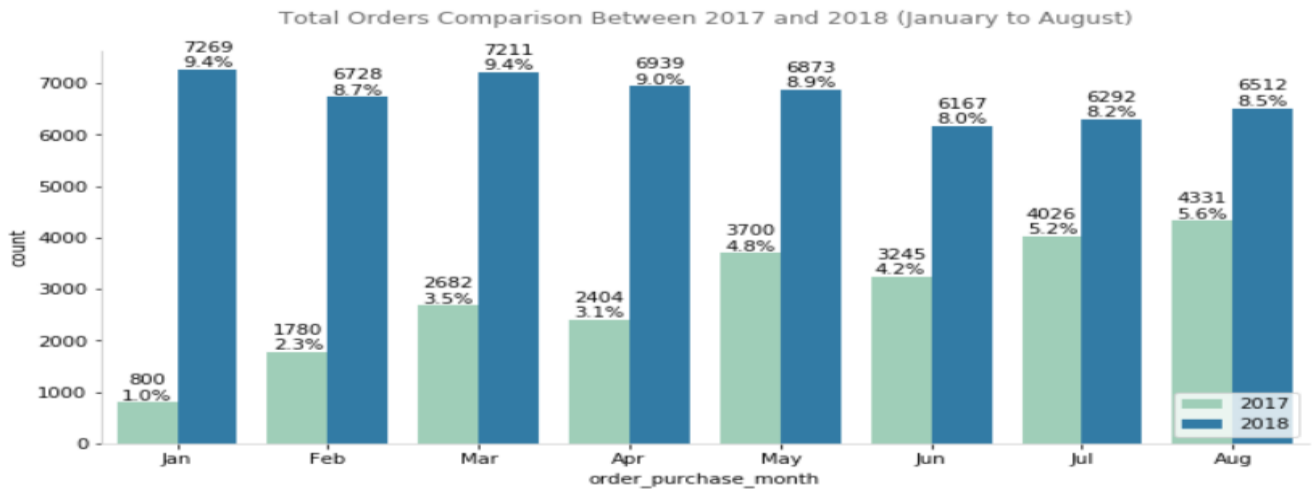
```
with CTE2017 as
(
 select
    sum(p.payment_value) as total_cost_of_order_2017,
    extract(month from o.order_purchase_timestamp) as Month,
 from Target_analysis.orders as o
 join Target_analysis.payments as p
 on o.order_id=p.order_id
 where extract(month from o.order_purchase_timestamp) between 1 and 8
 and extract(year from o.order_purchase_timestamp) = 2017
 group by  extract(month from o.order_purchase_timestamp)
),
 CTE2018 as
(
 select
    sum(p.payment_value) as total_cost_of_order_2018,
    extract(month from o.order_purchase_timestamp) as Month,
 from Target_analysis.orders as o
 join Target_analysis.payments as p
 on o.order_id=p.order_id
 where extract(month from o.order_purchase_timestamp) between 1 and 8
 and extract(year from o.order_purchase_timestamp) = 2018
 group by  extract(month from o.order_purchase_timestamp)
)
select
c1.month,
c1.total_cost_of_order_2017,
c2.total_cost_of_order_2018,
round((((c2.total_cost_of_order_2018-c1.total_cost_of_order_2017)/c1.total_cost_of_order_2017)*100),2) as percentage_increase

from CTE2017 as c1 join CTE2018 as c2
on c1.month=c2.month
order by month;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	month	total_cost_of_order_2017	total_cost_of_order_2018	percentage_increase
1	1	138488.03999999989	1115004.1800000065	705.13
2	2	291908.00999999966	992463.34000000334	239.99
3	3	449863.60000000027	1159652.1199999973	157.78
4	4	417788.03000000032	1160785.4800000025	177.84
5	5	592918.82000000111	1153982.1500000041	94.63
6	6	511276.38000000152	1023880.4999999946	100.26
7	7	592382.92000000284	1066540.7500000016	80.04
8	8	674396.32000000309	1022425.3199999979	51.61



Q 4 b) Mean & Sum of price and freight value by customer state

Query - Mean and sum of price

```
select
    c.customer_state,
    sum(oi.price) as Sum,
    avg(oi.price)as Mean
from Target_analysis.order_items as oi
join Target_analysis.orders as o on oi.order_id=o.order_id
join Target_analysis.Customers as c on c.customer_id= o.customer_id

group by c.customer_state
Order by mean
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	Sum	Mean
1	SP	5202955.05...	109.653629...
2	PR	683083.760...	119.004139...
3	RS	750304.020...	120.337453...
4	MG	1585308.02...	120.748574...
5	ES	275037.309...	121.913701...
6	SC	520553.340...	124.653577...
7	RJ	1824092.66...	125.117818...
8	DF	302603.939...	125.770548...
9	GO	294591.949...	126.271731...
10	BA	511349.990...	134.601208...

Query :- Mean and sum of freight value

```
select
    c.customer_state,
    sum(oi.freight_value) as Sum,
    avg(oi.freight_value)as Mean
from Target_analysis.order_items as oi
join Target_analysis.orders as o on oi.order_id=o.order_id
```

```
join Target_analysis.Customers as c on c.customer_id= o.customer_id

group by c.customer_state
Order by mean
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	Sum	Mean
1	SP	718723.069...	15.1472753...
2	PR	117851.680...	20.5316515...
3	MG	270853.460...	20.6301668...
4	RJ	305589.310...	20.9609239...
5	DF	50625.4999...	21.0413549...
6	SC	89660.2600...	21.4703687...
7	RS	135522.740...	21.7358043...
8	ES	49764.5999...	22.0587765...
9	GO	53114.9799...	22.7668152...
10	MS	19144.0300...	23.3748840...

Q5. Analysis on sales, freight and delivery time

a) Calculate days between purchasing, delivering and estimated delivery

Query:-

```
select
   order_purchase_timestamp,
   order_estimated_delivery_date,
   order_delivered_customer_date,
   timestamp_diff (order_estimated_delivery_date,order_purchase_timestamp ,day) as Estimated_days,
   timestamp_diff (order_delivered_customer_date,order_purchase_timestamp ,day) as Actual_days,
from Target_analysis.orders
where timestamp_diff (order_delivered_customer_date,order_purchase_timestamp ,day) is not null;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	Estimated_days	Actual_days
1	2016-10-07 14:52:30 UTC	2016-11-29 00:00:00 UTC	2016-10-14 15:07:11 UTC	52	7
2	2016-10-09 15:39:56 UTC	2016-12-08 00:00:00 UTC	2016-11-09 14:53:50 UTC	59	30
3	2016-10-09 00:56:52 UTC	2016-11-30 00:00:00 UTC	2016-10-16 14:36:59 UTC	51	7
4	2016-10-08 20:17:50 UTC	2016-11-30 00:00:00 UTC	2016-10-19 18:47:43 UTC	52	10
5	2016-10-03 21:01:41 UTC	2016-11-25 00:00:00 UTC	2016-11-08 10:58:34 UTC	52	35
6	2017-03-17 15:56:47 UTC	2017-05-18 00:00:00 UTC	2017-04-07 13:14:56 UTC	61	20
7	2017-03-20 11:01:17 UTC	2017-05-18 00:00:00 UTC	2017-03-30 14:04:04 UTC	58	10
8	2017-03-21 13:38:25 UTC	2017-05-18 00:00:00 UTC	2017-04-18 13:52:43 UTC	57	28
9	2018-08-20 15:56:23 UTC	2018-10-04 00:00:00 UTC	2018-08-29 22:52:40 UTC	44	9
10	2018-08-12 18:14:29 UTC	2018-10-04 00:00:00 UTC	2018-08-23 02:08:44 UTC	52	10

Q 5 b) Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- **time_to_delivery = order_purchase_timestamp-order_delivered_customer_date**

- **diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date**

Query:-

```
select
 timestamp_diff (order_delivered_customer_date,order_purchase_timestamp ,day) as Time_to_delivery,
 timestamp_diff (order_delivered_customer_date,order_estimated_delivery_date ,day) as Diff_estimated_delivery,
from Target_analysis.orders
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	Time_to_delivery	Diff_estimated_delivery
1	30	12
2	30	-28
3	35	-16
4	30	-1
5	32	0
6	29	-1
7	43	4
8	40	4
9	37	1
10	33	5

Q 5 c) Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Query:-

```
select
    c.customer_state,
    round(avg(freight_value),2)as Mean_freight_value,
    round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp ,day)),2) as Mean_time_to_delivery,
    round(avg(timestamp_diff(order_estimated_delivery_date ,order_delivered_customer_date,day)),2) as Mean_diff_in_estimated
_time
from Target_analysis.order_items as oi
join Target_analysis.orders as o on oi.order_id=o.order_id
join Target_analysis.Customers as c on o.customer_id=c.customer_id
group by c.customer_state
order by avg(freight_value);
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	customer_state	Mean_freight_value	Mean_time_to_delivery	Mean_diff_in_estimated_time
1	SP	15.15	8.26	10.27
2	PR	20.53	11.48	12.53
3	MG	20.63	11.52	12.4
4	RJ	20.96	14.69	11.14
5	DF	21.04	12.5	11.27
6	SC	21.47	14.52	10.67
7	RS	21.74	14.71	13.2
8	ES	22.06	15.19	9.77
9	GO	22.77	14.95	11.37
10	MS	23.37	15.11	10.34

Q 5 d) Sort the data to get the following:

a) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Query:-

```
select
    c.customer_state,
    round(avg(freight_value),2) as Avg_freight
from Target_analysis.order_items as oi
left join Target_analysis.orders as o on oi.order_id=o.order_id
left join Target_analysis.Customers as c on o.customer_id=c.customer_id
group by c.customer_state
order by avg(freight_value) desc
limit 5;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

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

b) Top 5 states with highest/lowest average time to delivery

Query:-

```
select
c.customer_state,
round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp ,day))) as Mean_time_to_delivery
from Target_analysis.order_items as oi
join Target_analysis.orders as o on oi.order_id=o.order_id
join Target_analysis.Customers as c on o.customer_id=c.customer_id
group by c.customer_state
order by avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp ,day)) desc
limit 5
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	Mean_time_to_delivery
1	RR	28.0
2	AP	28.0
3	AM	26.0
4	AL	24.0
5	PA	23.0

c) Top 5 states where delivery is really fast/ not so fast compared to estimated date

Query:-

```
select
c.customer_state,
round(avg(timestamp_diff(order_estimated_delivery_date ,order_delivered_customer_date,day))) as Mean_time_to_delivery
from Target_analysis.order_items as oi
join Target_analysis.orders as o on oi.order_id=o.order_id
join Target_analysis.Customers as c on o.customer_id=c.customer_id
group by c.customer_state
order by avg(timestamp_diff(order_estimated_delivery_date ,order_delivered_customer_date,day)) desc
limit 5
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	Mean_time_to_delivery
1	AC	20.0
2	RO	19.0
3	AM	19.0
4	AP	17.0
5	RR	17.0

Q 6) Payment type analysis:

a) Month over Month count of orders for different payment types

Query:-

```
select
  count(o.order_id) as Total_orders,
  avg(extract(Month from o.order_purchase_timestamp))as Month,
  p.payment_type

from Target_analysis.payments as p
join Target_analysis.orders as o on p.order_id=o.order_id
group by  extract(Month from o.order_purchase_timestamp), p.payment_type
order by Month,p.payment_type;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	Total_orders	Month	payment_type
1	1715	1.0	UPI
2	6103	1.0	credit_card
3	118	1.0	debit_card
4	477	1.0	voucher
5	1723	2.0	UPI
6	6609	2.0	credit_card
7	82	2.0	debit_card
8	424	2.0	voucher
9	1942	3.0	UPI
10	7707	3.0	credit_card

Results per page: 50 1 – 50 of 50

b) Count of orders based on the no. of payment installments

Query:-

```
select
    count(o.order_id)as Total_orders,
    p.payment_installments

from Target_analysis.payments as p
join Target_analysis.orders as o on p.order_id=o.order_id
group by p.payment_installments
order by p.payment_installments;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	Total_orders	payment_installments
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8
10	644	9

Insights and Recordation's:-

1) Time period of the data provided was observed between: Sept 2016 to October 2018
2) For Question 2 By the chart above we can conclude:

- E-commerce on Brazil really has a growing trend along the time.
- We can see some seasonality with peaks at specific months, but in general we can see clear that customers are more prone to buy things online than before.
- They tend to buy more at afternoons followed by Night followed by Morning which is followed by the Dawn.
- We have a sharp decrease between August 2018 and September 2018 and maybe the origin of that is related to noise/ any fault on data. For further comparison between 2017 and 2018, let's just consider orders between January and August in both years.

3) In this we analyzed the e-commerce orders variations month on month orders by state, and found that the SP state of Brazil had around 2 to 3 times more orders than average Brazil's orders.
4) Also, the analysis showed more orders in the month between October to February than the rest of the years.
5) From 4 we can draw insight that there was clear increase in the number of orders when we compare same month on month between 2017 and 2018, with a average increase of 200% meaning, order approximately became thrice in 2018 compared to 2018.

6) When compared mean and sum of price – It was very interesting to see how some states have a high total amount sold and a low price per order. If we look at SP (São Paulo) for example, it's possible to see that it is the state with most valuable state for e-commerce (5,188,099 sold) but it is also where customers pay less per order (110.00 per order).

7) When compared mean and sum of freight - we can get insights about the customers states with highest mean freight value. For example, customers in Roraima (RR), Paraiba (PB), Rondônia (RO) and Acre (AC) normally pays more than anyone on freights.
8) Insights with question 5 D
    - State RR with highest freight Value of 42.98
    - State SP with lowest freight Value of 15.15
    - State RR with maximum average time to delivery of 28 days
    - State SP with lowest average time to delivery of 8 days
9) Recommendation for the company is they need to deploy more delivery partners at the RR state and following 6-7 states where they are receiving most orders and due to high freight value delivery commute is high around 1 month. Which may result in the loss of customers.
10) In Question 6- we can see that payments made by credit card really took majority place on Brazilian e-commerce. Besides that, since 2018 march it's possible to see a little decrease on this type of payment. By the other side, payments made by debit card is showing a growing trend since 2018 may, which is a good opportunity for investor to improve services for payments like this.
11) By the analysis in question 6B - we can see how Brazilian customers prefer to pay the orders: mostly of them pay once into 1 installment and it's worth to point out the quantity of payments done by 10 installments.