

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

1.1 Data type of all columns in the "customers" table?

2

3 `select COLUMN_NAME, DATA_TYPE`

4 `from tactical-attic-402818.target.INFORMATION_SCHEMA.COLUMNS`

5 `where table_name = 'customers'`

### Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXE
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				

Or use schema option in big query

The screenshot shows the BigQuery interface. On the left, a sidebar lists workspace resources under the project 'tactical-attic-402818'. The 'target' dataset is expanded, showing the 'customers' table selected. The main panel displays the 'SCHEMA' tab for the 'customers' table. It includes a table with columns: Field name, Type, Mode, Key, Collation, Default Value, Policy Tags, and Description. The fields listed are customer\_id (STRING, NULLABLE), customer\_unique\_id (STRING, NULLABLE), customer\_zip\_code\_prefix (INTEGER, NULLABLE), customer\_city (STRING, NULLABLE), and customer\_state (STRING, NULLABLE). At the bottom, there are buttons for 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'.

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
customer_id	STRING	NULLABLE					
customer_unique_id	STRING	NULLABLE					
customer_zip_code_prefix	INTEGER	NULLABLE					
customer_city	STRING	NULLABLE					
customer_state	STRING	NULLABLE					

1.2 Get the time range between which the orders were placed ?

```

#order placed time is available in orders table i.e. which is
order_purchase_timestamp
#range- minimum value to maximum value
#using aggregate function for finding min() and max()
select min(order_purchase_timestamp) as starting_orderPlaced_time
,max(order_purchase_timestamp) as last_orderPlaced_time

from `target.orders`;

```

## Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON
Row	starting_orderPlaced_time ▼	last_orderPlaced_time ▼			
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC			

Insight : The orders are placed between September 26<sup>th</sup> 2016 to October 17<sup>th</sup> 2018

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

```
#Count the Cities & States of customers who ordered during the given period.  
#first find who are the customers are ordered between 2016 sep,26 to 2018 oct,17  
#for that use inner join it will fetch who are placed orders in the given period  
#use aggregated function count() on distinct city and state to find out from how  
many no of city and states the orders are placed  
select count(distinct customer_city) as City_count, count(distinct customer_state)  
as State_count  
from `target.customers` c  
inner join `target.orders` o  
on c.customer_id=o.customer_id;
```

#### Query results

JOB INFORMATION		RESULTS	CHART
Row	City_count ▼	State_count ▼	
1	4119	27	


Insights : The orders are placed from 4119 cities from 27 states between September 26<sup>th</sup> 2016 to October 17<sup>th</sup> 2018.

## 2.In-depth Exploration:

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

```
SELECT
  EXTRACT(year
FROM
  order_purchase_timestamp) AS Year,
  EXTRACT(month
FROM
  order_purchase_timestamp) AS Month,
  COUNT(order_id) AS No_of_orders_placed
FROM
  `target.orders`
GROUP BY
  1, 2
ORDER BY 1,2;
```

Query results

 SAVE RESULTS ▾

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

Row	Year ▾	Month ▾	No_of_orders_placed ▾
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

Insights : Yes there is positive increasing in no of orders placed 2016 to 2018.

Q2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
SELECT
  EXTRACT(month FROM order_purchase_timestamp) AS Month,
  COUNT(order_id) AS No_of_orders_placed
FROM `target.orders`
GROUP BY 1
ORDER BY 1;
```

### Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON
Row	Month ▼	No_of_orders_placed ▼			
1	1	8069			
2	2	8508			
3	3	9893			
4	4	9343			
5	5	10573			
6	6	9412			
7	7	10318			
8	8	10843			
9	9	4305			
10	10	4959			

Insights : we observe from January to August the no of orders are increasing. then in September and October the orders are comes down from their onwards again the no of orders are increasing(oct to dec).


Recommendation : Plan any discounts or offers in off peak seasons(like September,October) to boost the no of orders.

2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

```
select case
```

```
when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Mornings'
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 time_of_day, count(order_id) as count_of_orders
from `target.orders`
group by time_of_day
order by count_of_orders desc;
```

Query results				 SAVE RESULTS ▼
<	JOB INFORMATION	RESULTS	CHART	PREVIEW
Row	time_of_day ▼	count_of_orders ▼		
1	Afternoon	38135		
2	Night	28331		
3	Mornings	27733		
4	Dawn	5242		

Insight: The Brazilian Customers mostly place their orders in Afternoon time around 1 p.m. to 6p.m.This is a peak period ,with 38K+ orders placed, which is highest of the day.

Recommendations : Giving Special discounts in this peak period attracts even more orders.

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

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

```
select customer_state,  
extract(month from order_purchase_timestamp) as month,  
count(order_id) as no_of_orders  
from `target.orders` o  
inner join `target.customers` c  
on o.customer_id = c.customer_id  
group by 1,2  
order by 1,2;
```

Query results					SAVE RESULTS	EXPLORE DATA	
<	JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAIL	>
Row	customer_state	month	no_of_orders				
1	AC	1	8				
2	AC	2	6				
3	AC	3	4				
4	AC	4	9				
5	AC	5	10				
6	AC	6	7				
7	AC	7	9				
8	AC	8	7				
9	AC	9	5				
10	AC	10	6				
Results per page:					50	1 – 50 of 322	< < > >

Insights : The state SP have highest number of orders compare to remaining states in every month and states like RR,AC,AM,AP are having least no of orders which are less then or equal to 10

Recommendations : Implement brand loyalty programmes in states like SP to retain loyal customers , try to increase service to states like RR,AC,AM,AP to boost the count of orders placed.

### 3.2 How are the customers distributed across all the states?

#How are the customers distributed across all the states?

```
select customer_state, count(customer_id) as count_of_customers,
concat(round(
(count(customer_id)/(select count(customer_id) from `target.customers`))*100
),'%') as customer_distribution
from `target.customers`
group by customer_state
order by count_of_customers desc;
```

Query results

 SAVE RESULTS ▾

 EXPLORE DATA ▾



<	JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAIL	>
Row	customer_state ▾	count_of_customers	customer_distribution ▾				
1	SP	41746	42%				
2	RJ	12852	13%				
3	MG	11635	12%				
4	RS	5466	5%				
5	PR	5045	5%				
6	SC	3637	4%				
7	BA	3380	3%				
8	DF	2140	2%				
9	ES	2033	2%				
10	GO	2020	2%				

Insights : The no of customers are mostly from SP state around 41K+ .  
In the whole Brazil ,42 % of customer distribution is from SP state.



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

**4.1** Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

#cte to find out cost of orders for year and month wise i.e order\_cost cte

```
with Order_cost as(
select year,month,sum(cost_of_order) as cost_of_order
from
(
select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
pt.order_id,pt.payment_value as cost_of_order
from `target.payments` pt
inner join `target.orders` ot
on pt.order_id=ot.order_id
)temp
where year in (2017,2018) and (month between 1 and 8)
group by year,month
order by year,month
)
select '2017-2018' as year ,month,cost_of_order_2017,cost_of_order_2018,
round((((cost_of_order_2018-cost_of_order_2017)/cost_of_order_2017)*100)as
Percentage_increase
from
(select year,month,cost_of_order as
cost_of_order_2017,lead(cost_of_order,8)over(order by year,month) as
cost_of_order_2018 from order_cost
)t
```

Query results

[SAVE RESULTS](#)

[EXPLOR](#)

<	JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION I
Row	year	month	cost_of_order_2017	cost_of_order_2018	Percentage_increase		
1	2017-2018	1	138488.0399999...	1115004.180000...	705.0		
2	2017-2018	2	291908.0099999...	992463.3400000...	240.0		
3	2017-2018	3	449863.6000000...	1159652.119999...	158.0		
4	2017-2018	4	417788.0300000...	1160785.479999...	178.0		
5	2017-2018	5	592918.8200000...	1153982.149999...	95.0		
6	2017-2018	6	511276.3800000...	1023880.499999...	100.0		
7	2017-2018	7	592382.9200000...	1066540.750000...	80.0		
8	2017-2018	8	674396.3200000...	1022425.320000...	52.0		

Insights : From 2017-2018(Jan to Aug) the percentage increase in cost\_of\_orders. We seen that January month has highest %increase which is 705% which means the cost of orders increased by 7.05 times then previous year. (i.e.  $138488 + (138488 * 7.05) \sim 1115004$ ) and the least %increase seen in August month 52% only

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

```
select customer_state ,
round(sum(price))as total_order_value,
round(avg(price)) as average_order_value
from `target.order_items` oit
inner join `tactical-attic-402818.target.orders` ot
on ot.order_id=oit.order_id
inner join `target.customers` ct
on ct.customer_id = ot.customer_id
group by customer_state;
```

Query results

SAVE RESULTS

EX

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

EXECUT

Row	customer_state	total_order_value	average_order_value
1	SP	5202955.0	110.0
2	RJ	1824093.0	125.0
3	PR	683084.0	119.0
4	SC	520553.0	125.0
5	DF	302604.0	126.0
6	MG	1585308.0	121.0
7	PA	178948.0	166.0
8	BA	511350.0	135.0
9	GO	294592.0	126.0
10	RS	750304.0	120.0

Results per page:

50

1 – 27 of 27

Results per page: 50 1 – 27 of 27

Insights : top 3 total order values from SP,RJ,MG respectively and average order value of these states are 110,125,121 because these are top 3 most populated states in Brazil and top 3 least total order values from RR,AP,AC because these are top 3 least populated states in Brazil.

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

```
select customer_state ,
round(sum(freight_value))as total_freight_value,
round(avg(freight_value)) as average_freight_value
from `target.order_items` oit
inner join `tactical-attic-402818.target.orders` ot
on ot.order_id=oit.order_id
inner join `target.customers` ct
on ct.customer_id = ot.customer_id
group by customer_state
order by 2 desc;
```

#### Query results

[SAVE RESULTS](#)

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	customer_state	total_freight_value	average_freight_valu			
1	SP	718723.0	15.0			
2	RJ	305589.0	21.0			
3	MG	270853.0	21.0			
4	RS	135523.0	22.0			
5	PR	117852.0	21.0			
6	BA	100157.0	26.0			
7	SC	89660.0	21.0			
8	PE	59450.0	33.0			
9	GO	53115.0	23.0			
10	DF	50625.0	21.0			



Insights : The Average freight values are in the range from 15 to 43.the highest total freight value from SP state with average freight value as 15 and lowest total freight value from RR state with average freight value as 43.freight value indicate transportation charge by this we can say the average transportation charge is nearly 3 times higher in RR state compared to SP state .

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

### Q.5.1

Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

```
select datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver_in_days,
datetime_diff(order_estimated_delivery_date ,order_delivered_customer_date,day) as
diff_estimated_delivery_in_days
from
`target.orders`
where order_estimated_delivery_date is not null and order_delivered_customer_date
is not null
order by order_purchase_timestamp;
```

Query results				 SAVE RESULTS ▾	 E
JOB INFORMATION				RESULTS	CHART
				PREVIEW	JSON
				EXECUTION DETAILS	EXECU
Row	time_to_deliver_in_days ▾	diff_estimated_delivery_in_days ▾			
1	54	-36			
2	23	0			
3	24	10			
4	35	16			
5	30	25			
6	27	22			
7	10	39			
8	30	27			
9	28	23			
10	18	32			

Results per page: 50 ▾ 1 – 50 of 96476

Insights : Most of the time the customer received delivery date earlier than customer estimated delivery date . negative number represent that customer some time receives delivery date longer than customer estimated delivery date .

Recommendations : try to decrease the cases of such negative days (i.e. delivery date longer than customer estimated delivery date)

## Q5.2

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

```
with rank_final as (
    select customer_state ,
           round(avg(freight_value)) as average_freight_value,
           dense_rank()over(order by round(avg(freight_value))desc)as rank_least,
           dense_rank()over(order by round(avg(freight_value)) asc) as rank_top
    from `target.order_items` oit
    inner join `tactical-attic-402818.target.orders` ot
    on ot.order_id=oit.order_id
    inner join `target.customers` ct
    on ct.customer_id = ot.customer_id
    group by customer_state
)

select customer_state ,
       average_freight_value,
       case
         when rank_least <=5 then concat('Highest',' ',rank_least)
         when rank_top <=5 then concat('Lowest',' ',rank_top)
       end as top5
from rank_final
where rank_least <=5 or rank_top <=5
order by top5,rank_final.average_freight_value ;
```

Query results

SAVE RESULTS

EXPORT

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

EXECUTION TIME

Row	customer_state	average_freight_value	top5	
1	RR	43.0	Highest 1	
2	PB	43.0	Highest 1	
3	RO	41.0	Highest 2	
4	AC	40.0	Highest 3	
5	PI	39.0	Highest 4	
6	MA	38.0	Highest 5	
7	SP	15.0	Lowest 1	
8	RJ	21.0	Lowest 2	
9	MG	21.0	Lowest 2	
10	PR	21.0	Lowest 2	

Results per page:

50

1 – 17 of 17

Insights: SP state has lowest average freight value ,RR state has the highest average freight value. The top5 lowest freight and top5 highest freight values are listed in above table.

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

```
with final as (
select customer_state,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver_in_days
from
`target.orders` ot
inner join `target.customers` ct
on ct.customer_id=ot.customer_id
where order_delivered_customer_date is not null
order by order_purchase_timestamp),
  final_1 as
(select customer_state,
round(avg(final.time_to_deliver_in_days)) as average_delivery_time_in_days
from final
group by 1),
rank_top5 as(
select customer_state, average_delivery_time_in_days,
dense_rank()over(order by final_1.average_delivery_time_in_days desc) as rank_top,
dense_rank()over(order by final_1.average_delivery_time_in_days asc) as rank_least
from final_1)
select customer_state, average_delivery_time_in_days,case
when rank_top<=5 then concat('Highest',' ',rank_top)
when rank_least<=5 then concat('Lowest',' ',rank_least)
end as top5
from rank_top5
where rank_top <=5 or rank_least<=5
order by top5
```

Query results <a href="#">SAVE RESULTS</a>				
JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	customer_state	average_delivery_tim	top5	
1	RR	29.0	Highest 1	
2	AP	27.0	Highest 2	
3	AM	26.0	Highest 3	
4	AL	24.0	Highest 4	
5	PA	23.0	Highest 5	
6	SP	8.0	Lowest 1	
7	MG	12.0	Lowest 2	
8	PR	12.0	Lowest 2	
9	DF	13.0	Lowest 3	
10	SC	14.0	Lowest 4	

Insights : top 5 highest average delivery time taking states are RR,AP,AM,AL,PA and the top5 lowest average delivery time taking states are SP,MG,PR,DF,SC etc.

Recommendations: Enhance Logistic and Shipping services to reduce the delivery time thus increase customer satisfaction even better.

#### Q5.4

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.

WITH

```
order_delivery AS (
SELECT
    customer_state,
    ROUND(AVG(DATETIME_DIFF(order_estimated_delivery_date,order_delivered_customer_
date,day))) AS
    average_order_delivery_in_days
FROM `target.orders` ot
INNER JOIN `target.customers` ct
ON ct.customer_id= ot.customer_id
WHERE
    order_estimated_delivery_date IS NOT NULL AND order_delivered_customer_date IS
NOT NULL
GROUP BY customer_state)
SELECT customer_state,average_order_delivery_in_days,fast_delivery_rank
FROM (SELECT customer_state,average_order_delivery_in_days,
    DENSE_RANK()OVER(ORDER BY order_delivery.average_order_delivery_in_days) AS
fast_delivery_rank FROM order_delivery )TEMP
WHERE fast_delivery_rank <=5
ORDER BY 2 ;
```

Query results

SAVE RESULTS

EX

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

EXECUT

Row	customer_state	average_order_delivery_in_days	fast_delivery_rank
1	AL	8.0	1
2	MA	9.0	2
3	SE	9.0	2
4	BA	10.0	3
5	CE	10.0	3
6	ES	10.0	3
7	PI	10.0	3
8	MS	10.0	3
9	SP	10.0	3
10	TO	11.0	4

Results per page: 50

1 – 18 of 18

PERSONAL HISTORY

PROJECT HISTORY

Insights : there are 18 states out of 27 states which are in top 5 for fast delivery compared to estimated date delivery. AL state Tops the table where order delivers really fast then the estimated delivery date.

## Q6 Analysis based on the payments:

### Q6.1

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

```
select payment_type,format_date('%b',date(1,months,1)) as month ,no_of_orders
from
(select payment_type,extract(month from order_purchase_timestamp) as
months,count(distinct pt.order_id) as no_of_orders
from `target.payments` pt
inner join `target.orders` ot
on ot.order_id=pt.order_id
group by payment_type,months) temp

order by payment_type,months;
```

Query results

SAVE RESULTS

EX

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

EXECUT

Row	payment_type	month	no_of_orders	
1	UPI	Jan	1715	
2	UPI	Feb	1723	
3	UPI	Mar	1942	
4	UPI	Apr	1783	
5	UPI	May	2035	
6	UPI	Jun	1807	
7	UPI	Jul	2074	
8	UPI	Aug	2077	
9	UPI	Sep	903	
10	UPI	Oct	1056	

Results per page:

50

1 – 50 of 50

Insights: They are 4 types of payment types Credit Card , UPI , Debit Card , Vouchers among these most no of orders placed through Credit Card so Credit card is the most popular payment type . least no\_of\_orders\_placed through vouchers payment type. August having most no of orders placed in all the type of payment modes and September having least no of orders placed in all the type of payment modes.



Q6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

```
select distinct payment_installments , count(distinct order_id) as no_of_orders
from `target.payments`
group by payment_installments
order by payment_installments;
```

### Query results

Job Information

Results

Chart

Preview

JSON

Row	payment_installments	no_of_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644

Results

Insights : The highest no. of orders are placed on basis of one payment installment. The maximum no. of payment installment is 24 having only 18 orders.