

Scaler Business case

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

```
select column_name,data_type from red-archive-396816.Target_SQL.INFORMATION_SCHEMA.COLUMNS where table_name='customers';
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
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		

Insights

Customers wants to check the data types of all columns in their table. I create a table name using scheme and with Data type() command I used to find the data types for all the columns.

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

```
select min(order_purchase_timestamp) as minimum, max(order_purchase_timestamp) as maximum, from Target_SQL.orders;
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	minimum	maximum					
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC					

Insights

Customers order placed by the min time range from 2016 to maximum time range till 2018.

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

```
SELECT CONCAT(customer_city, ', ', customer_state) AS location, COUNT(*) AS count
FROM Target_SQL.customers
GROUP BY location;
```

JOB INFORMATION		RESULTS	JSON	EXECUTIO
Row	location	count		
11	una, BA	5		
12	anta, RJ	4		
13	avai, SP	1		
14	bage, RS	65		
15	bodo, RN	1		
16	bora, SP	1		
17	buri, SP	10		
18	cacu, GO	8		
19	caem, RA	1		

Insights: Counting that the customers who placed their order from different place

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

```
select count(order_id) as cou_ord, extract(year from
order_purchase_timestamp) as year from Target_SQL.orders
group by year
order by year;
```

Query results

JOB INFORMATION		RESULTS	JSON	E
Row	cou_ord	year		
1	329	2016		
2	45101	2017		
3	54011	2018		

Insights – In 2016 the order is 329 and it is upright upto 54011 in the year 2018.

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

```
select count(order_id) as NoofOrders, extract(month from
order_purchase_timestamp) as month, extract(year from
```

```

order_purchase_timestamp) as year,count(order_id) /
(sum(count('NoofOrders')) OVER( order by 'year' ))*100 as percent
from Target_SQL.orders
group by month,year
order by month,year;

```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW
Row	NoofOrders	month	year	percent		
1	800	1	2017	0.804497139007...		
2	7269	1	2018	7.309862129302...		
3	1780	2	2017	1.790006134290...		
4	6728	2	2018	6.765820939049...		

Insights – The number of orders are placed in each month and year with significant data.

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

```

with dt as(select count(o.order_id) as totalorders ,"DAWN" as time
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
EXTRACT(HOUR FROM o.order_purchase_timestamp) between 0 and 6
union all
select count(o.order_id) as totalorders,"MRN" as time
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
EXTRACT(HOUR FROM o.order_purchase_timestamp) between 7 and 12
union all
select count(o.order_id) as totalorders,"AFT" as time
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
EXTRACT(HOUR FROM o.order_purchase_timestamp) between 13 and 18
union all

```

```

select count(o.order_id) as totalorders, "NIGHT" as time
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
EXTRACT(HOUR FROM o.order_purchase_timestamp) between 18 and 23)
select * from dt
order by totalorders desc;

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTIC
Row	totalorders	time		
1	16104	AFT		
2	14098	NIGHT		
3	11664	MRN		
4	2258	DAWN		

Insights – Brazilian customers purchasing from range 16104 in the Afternoon is the maximum orders while in Dawn the orders will around 2258.

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

```

select count(o.order_id) as totalorders, EXTRACT(month FROM
o.order_purchase_timestamp) as month, EXTRACT(year FROM
o.order_purchase_timestamp) as year, c.customer_state
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
group by month,year,c.customer_state
order by month,year,c.customer_state;

```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART
Row	totalorders	month	year	customer_state	
1	2	1	2017	AC	
2	2	1	2017	AL	
3	25	1	2017	BA	
4	9	1	2017	CE	
5	13	1	2017	DF	
6	12	1	2017	ES	
7	18	1	2017	GO	
8	9	1	2017	MA	

Insights : All states orders are placed in the first month of the year 2017 and 2018.

3.2 How are the customers distributed across all the states?

```
select count(customer_unique_id) as totalCustomer, customer_state from
Target_SQL.customers
group by customer_state
order by totalCustomer desc;
```

JOB INFORMATION		RESULTS	JSON
Row	totalCustomer	customer_state	
1	41746	SP	
2	12852	RJ	
3	11635	MG	
4	5466	RS	
5	5045	PR	
6	3637	SC	
7	3380	BA	
8	2140	DF	

Insights - maximum 41746 number of customers in state SP

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

```
select Round(sum(p.payment_value),2) as totalordervalue,
round((avg(p.payment_value)),2) as stateavg, c.customer_state as statename
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
inner join Target_SQL.payments as p on o.order_id = p.order_id
group by c.customer_state;
```

Row	totalordervalue	stateavg	statename
1	2144379.69	158.53	RJ
2	890898.54	157.18	RS
3	5998226.96	137.5	SP
4	355141.08	161.13	DF
5	811156.38	154.15	PR
6	187029.29	195.23	MT
7	152523.02	198.86	MA
8	96962.06	227.08	AL

Insights – The total and average value is calculated with some of payment value for the each state.

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

```
select Round(sum(oi.freight_value),2) as totalfreightvalue,
round((avg(oi.freight_value)),2) as stateavg, c.customer_state as statename
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
inner join Target_SQL.order_items as oi on o.order_id = oi.order_id
group by c.customer_state;
```

JOB INFORMATION	RESULTS	JSON	EXECUTION DET/
Row	totalfreightvalue	stateavg	statename
1	29715.43	28.17	MT
2	31523.77	38.26	MA
3	15914.59	35.84	AL
4	718723.07	15.15	SP
5	270853.46	20.63	MG
6	59449.66	32.92	PE
7	305589.31	20.96	RJ
8	50625.5	21.04	DF

Insights – The total and average are calculated with freight value for each state.

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 date_diff(order_delivered_customer_date,
order_purchase_timestamp,day) as time_to_deliver ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date, day)
as diff_estimated_delivery, order_id
from Target_SQL.orders
limit 10;
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	time_to_deliver	diff_estimated_delivery	order_id	
1	null	null	7a4df5d8cff4090e541401a20a...	
2	null	null	35de4050331c6c644cddc86f4...	
3	null	null	b5359909123fa03c50bdb0cfe...	
4	null	null	dba5062fbda3af4fb6c33b1e04...	
5	null	null	90ab3e7d52544ec7bc3363c82...	
6	null	null	fa65dad1b0e818e3ccc5cb0e3...	
7	null	null	1df2775799eecd9dd8502425...	
8	null	null	6190c94657c1012992c274b8...	

Insights- It is calculating the time with estimated delivery time and displaying with order id.

5 . 2 Find out the top 5 states with the highest & lowest average freight value

```
with dt1 as (select Round(sum(oi.freight_value),2) as totalfreightvalue,
round((avg(oi.freight_value)),2) as stateavg, c.customer_state as statename
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
inner join Target_SQL.order_items as oi on o.order_id = oi.order_id
group by customer_state
order by stateavg
limit 5),
```

```

dt2 as (select Round(sum(oi.freight_value),2) as totalfreightvalue,
round((avg(oi.freight_value)),2) as stateavg, c.customer_state as statename
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
inner join Target_SQL.order_items as oi on o.order_id = oi.order_id
group by customer_state
order by stateavg desc
limit 5)

```

```

select * from dt1
union all
select * from dt2

```

Row	totalfreightvalue	stateavg	statename
1	718723.07	15.15	SP
2	117851.68	20.53	PR
3	270853.46	20.63	MG
4	305589.31	20.96	RJ
5	50625.5	21.04	DF
6	2235.19	42.98	RR
7	25719.73	42.72	PB
8	11417.38	41.07	RO
9	3686.75	40.07	AC
10	21218.2	39.15	PI

Insights - The maximum value is 718723.07 and state average is 15 out of top 5 states.

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

```

with dt1 as (select round(((sum(date_diff(order_delivered_customer_date,
order_purchase_timestamp,day))) /count(order_id)),2) as Avg_delivery_time,
c.customer_state as statename
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
group by customer_state
order by Avg_delivery_time
limit 5),
dt2 as (select round(((sum(date_diff(order_delivered_customer_date,
order_purchase_timestamp,day))) /count(order_id)),2) as Avg_delivery_time,
c.customer_state as statename
from Target_SQL.orders as o

```



```

inner join Target_SQL.customers as c on o.customer_id=c.customer_id
group by customer_state
order by Avg_delivery_time desc
limit 5)
select * from dt1
union all
select * from dt2

```

JOB INFORMATION		RESULTS	JSON
Row	Avg_delivery_time	statename	
1	8.05	SP	
2	11.25	PR	
3	11.27	MG	
4	12.16	DF	
5	14.12	SC	

Insights - The maximum Average delivery time value is 8.05 and state name is SP out of top 5 states.

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.

```

select round((sum(date_diff(order_delivered_customer_date,
order_purchase_timestamp,day)) /count(order_id)) -
(sum(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day))/count(order_id)),2) as fastest_delivery, customer_state
from Target_SQL.orders as o
inner join Target_SQL.customers as c on o.customer_id=c.customer_id
group by customer_state
order by fastest_delivery
limit 5;

```


JOB INFORMATION		RESULTS	JSON	EX
Row	fastest_delivery	customer_state		
1	-1.78	SP		
2	-0.82	PR		
3	-0.74	MG		
4	-0.21	RO		
5	0.86	AC		

Insights - The maximum fastest delivery is -1.78 of state is SP out of top five states.

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

```
select count(o.order_id) as NoofOrders,P.payment_type,extract(month from
order_purchase_timestamp) as month,extract(year from
order_purchase_timestamp) as year
from Target_SQL.orders as O
inner join Target_SQL.payments as P on O.order_id=P.order_id
group by month, year, payment_type
order by month, year, payment_type;
```

Query results

 SAVE RESULTS ▾

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

E

Row	NoofOrders	payment_type	month	year	
1	197	UPI	1	2017	
2	583	credit_card	1	2017	
3	9	debit_card	1	2017	
4	61	voucher	1	2017	
5	1518	UPI	1	2018	
6	5520	credit_card	1	2018	
7	109	debit_card	1	2018	

Insights : The different payment method is paid from Number of orders on each month.

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

```
select payment_installments,payment_sequential,count(*) as
Number_of_orders
from Target_SQL.orders o join Target_SQL.payments p on o.order_id =
p.order_id
where payment_installments>1
group by payment_installments,payment_sequential
order by Number_of_orders;
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	payment_installment	payment_sequential	Number_of_orders	
1	3	3	1	
2	23	1	1	
3	22	1	1	
4	21	1	3	
5	16	1	5	
6	7	2	7	
7	17	1	8	
8	14	1	15	

Insights – Calculated the no. of orders on the basis of the payment installments which the customers have paid.

OVERALL INSIGHTS

- There are few orders in the last quarter of the year good offers can be provided during that time to increase the sales.
- There are few states where the customer base is good but many states are having a number of customers which is low, strategies to attract them should be in place.
- Some states have average order price and freight high compared to others. Networking with delivery partners is recommended.
- The delivery time is high in most of the states. Reliable delivery partner collaboration is recommended.
- Order placed using credit cards and on EMI is very high. Company can try to bring offers on credit cards.