Target SQL Business Case

By

Aman Priyadarshi,

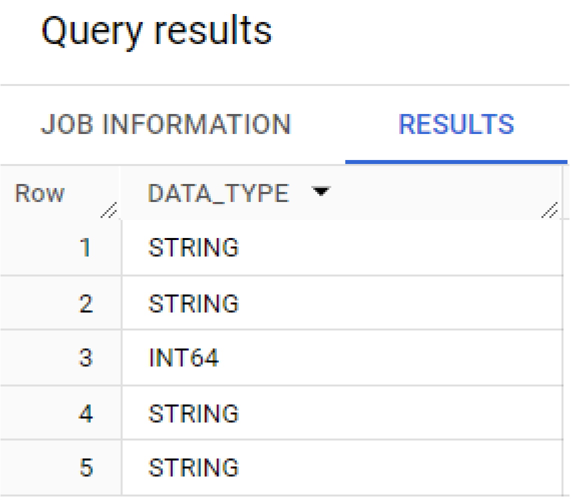
DSML Nov’23, Tuesday Morning Batch

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

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

Query –

```
SELECT DATA_TYPE
from target.INFORMATION_SCHEMA.COLUMNS
where table_schema = 'target' and table_name = 'customers'
```



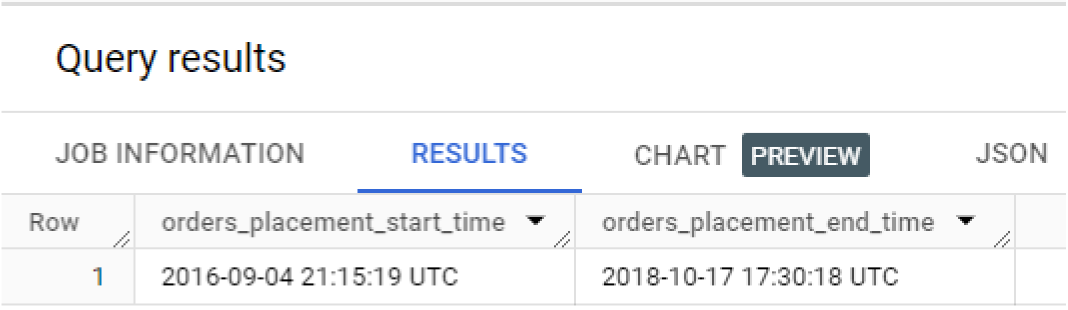
Row	DATA_TYPE
1	STRING
2	STRING
3	INT64
4	STRING
5	STRING

Insight: The query shows the data types of the columns of table ‘customers’ from the data set ‘target’

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

   Query –

```
select min(order_purchase_timestamp)
       as`orders_placement_start_time`,
       max(order_purchase_timestamp)
       as`orders_placement_end_time`
from target.orders
```



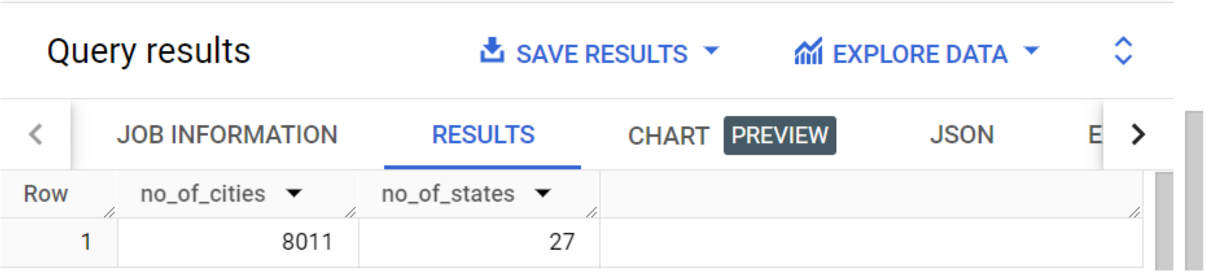
Row	orders_placement_start_time	orders_placement_end_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insight: The query shows the date and time of the placement of first order and that of the last order.

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

Query -

```
select count(distinct geolocation_city) as `no_of_cities`,
       count(distinct geolocation_state) as `no_of_states`
from target.geolocation
```



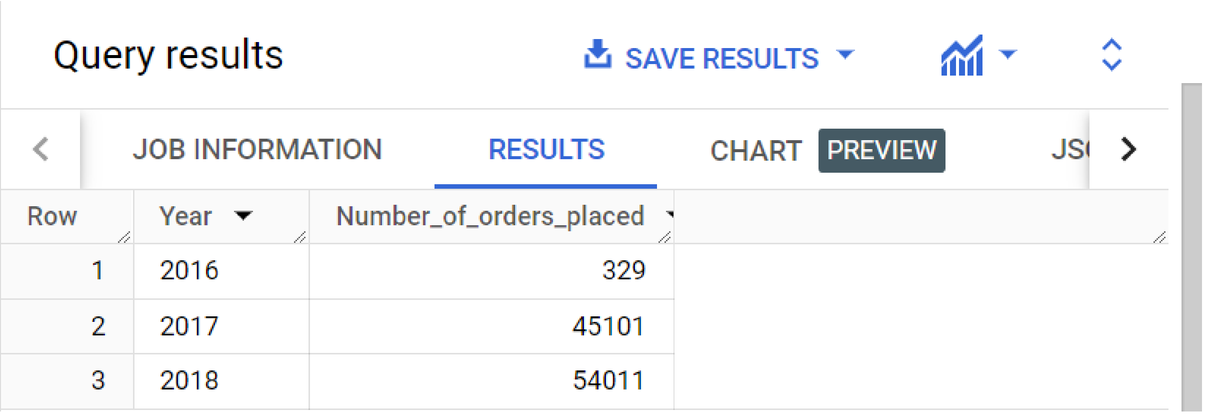
Row	no_of_cities	no_of_states
1	8011	27

Insight: The query shows that orders placed in the given time range were placed from 8011 different cities belonging to 27 different states.

In-depth Exploration:

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

```
select case
       when extract(year from order_purchase_timestamp)= 2016
       then '2016'
       when extract(year from order_purchase_timestamp)= 2017
       then '2017'
       when extract(year from order_purchase_timestamp)= 2018
       then '2018'
       end as `Year`,
       count(order_id) as `Number_of_orders_placed`
from `target.orders`
group by `Year`
order by `Number_of_orders_placed`
```



Row	Year	Number_of_orders_placed
1	2016	329
2	2017	45101
3	2018	54011

Insight: The query shows that there was a vast increase in the number of orders placed from the year 2016 to the year 2017 whereas the increment in the number of orders placed from the year 2017 to the year 2018 was also significant.

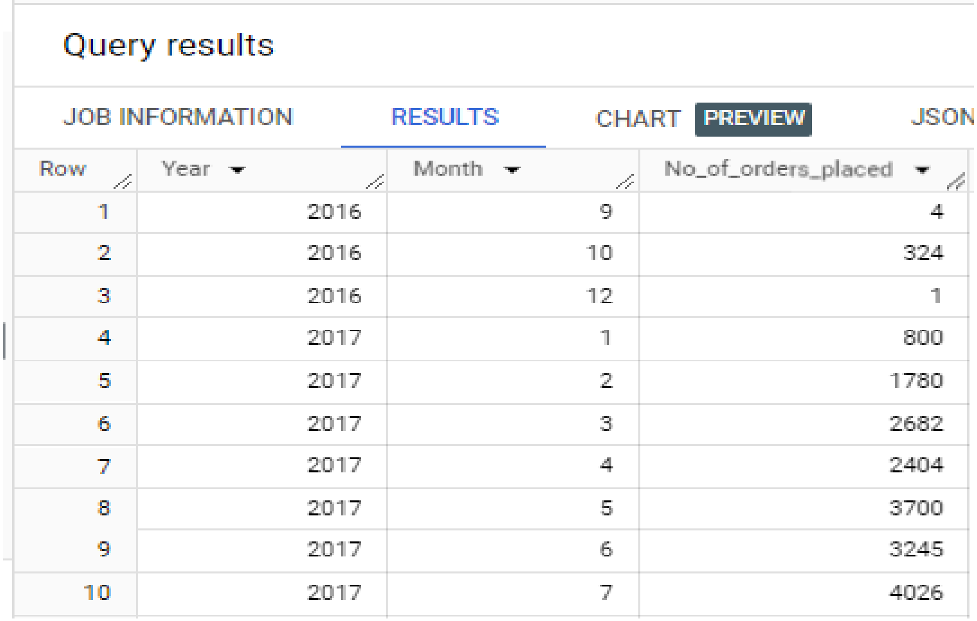
Recommendation: Following are the recommendations based on the insights of the query:

a. The stocking of products should be increased to meet the increasing number of orders,
b. The above needs to be supported by increased space for orders and better logistics facilities.

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

Query -

```
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 `Year`,`Month`
order by `Year`,`Month`
```



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

Insight: The query shows the seasonality in some months of the year. Most of it is evident in the mid months of the year namely, June, July and August. Second seasonality, in terms of sales figure, can be seen in the months of November and January due to them being festive season and the first month, respectively.

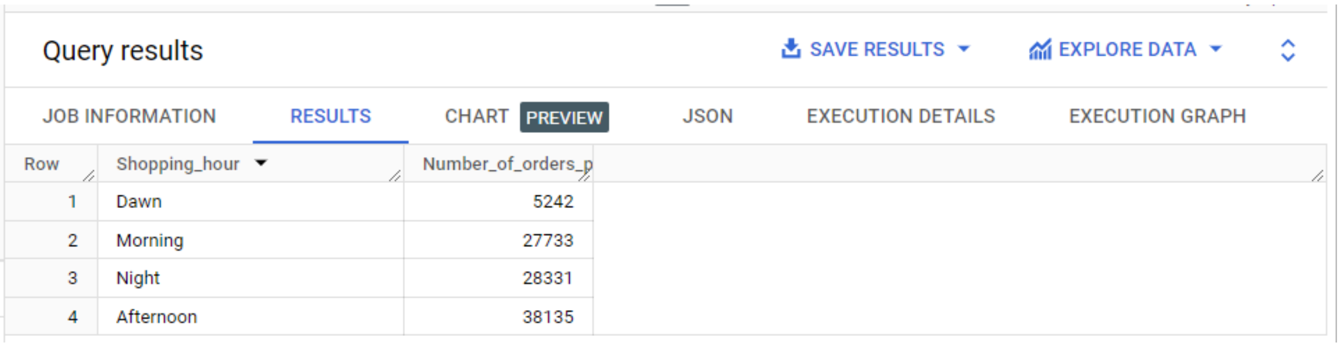
Recommendation: Following are the recommendations based on the insights of the query:

a. Special discounts or sale could be launched in these months,
b. Price drop or combo offers could be provided to the customers.

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

Query -

```
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 "Morning"
       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 `Shopping_hour`,
       count(order_id) as `Number_of_orders_placed`
from `target.orders`
group by `Shopping_hour`
order by `Number_of_orders_placed`
```



Row	Shopping_hour	Number_of_orders_p
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

Insights: The query shows that the Brazilian customers are the most active for shopping in the afternoon, followed by at night, morning and dawn. Collectively, a lot of people shops during afternoon and night.

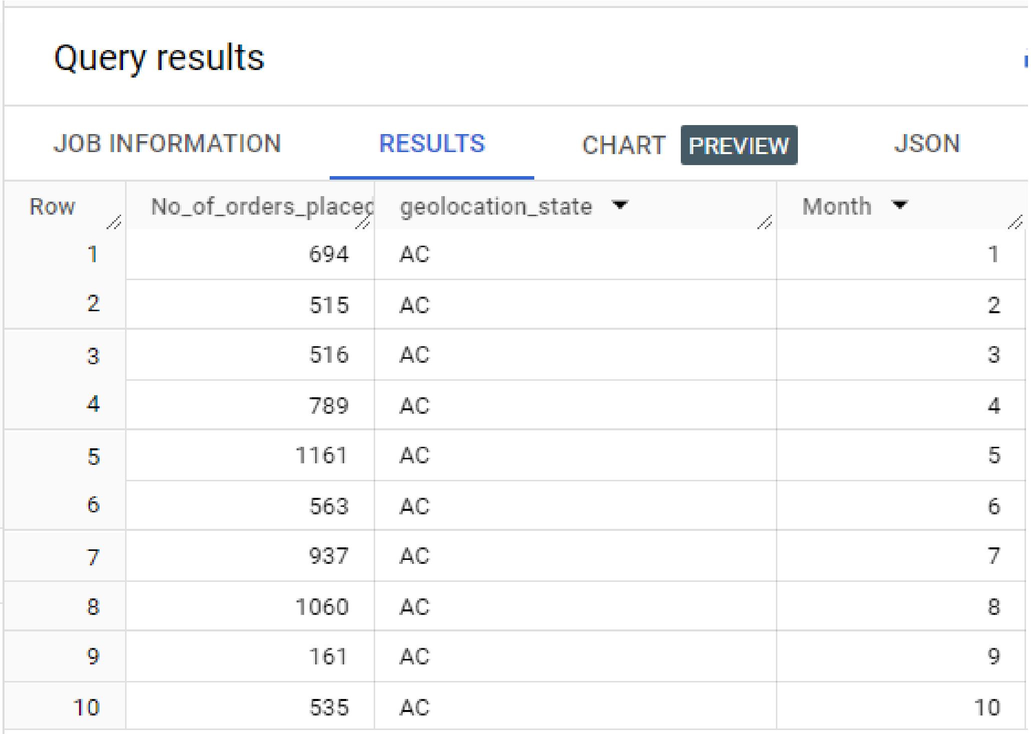
Recommendation: Following are the recommendations based on the insights of the query:

a. Attractive offers can be offered to the customers at these hours,
b. Hourly sale can be launched in between these hours,
c. More number of support staffs should be active for this time period.

Evolution of E-commerce orders in the Brazil region:

1. Get the month-on-month no. of orders placed in each state.
Query -

```
select count(o.order_id) as `No_of_orders_placed`,
       g.geolocation_state,
       extract(month from o.order_purchase_timestamp) as
       `Month`
from target.orders as o
inner join target.customers as c
on o.customer_id=c.customer_id
inner join target.geolocation as g
on c.customer_zip_code_prefix=g.geolocation_zip_code_prefix
group by g.geolocation_state,Month
order by g.geolocation_state,Month
```



Row	No_of_orders_placed	geolocation_state	Month
1	694	AC	1
2	515	AC	2
3	516	AC	3
4	789	AC	4
5	1161	AC	5
6	563	AC	6
7	937	AC	7
8	1060	AC	8
9	161	AC	9
10	535	AC	10

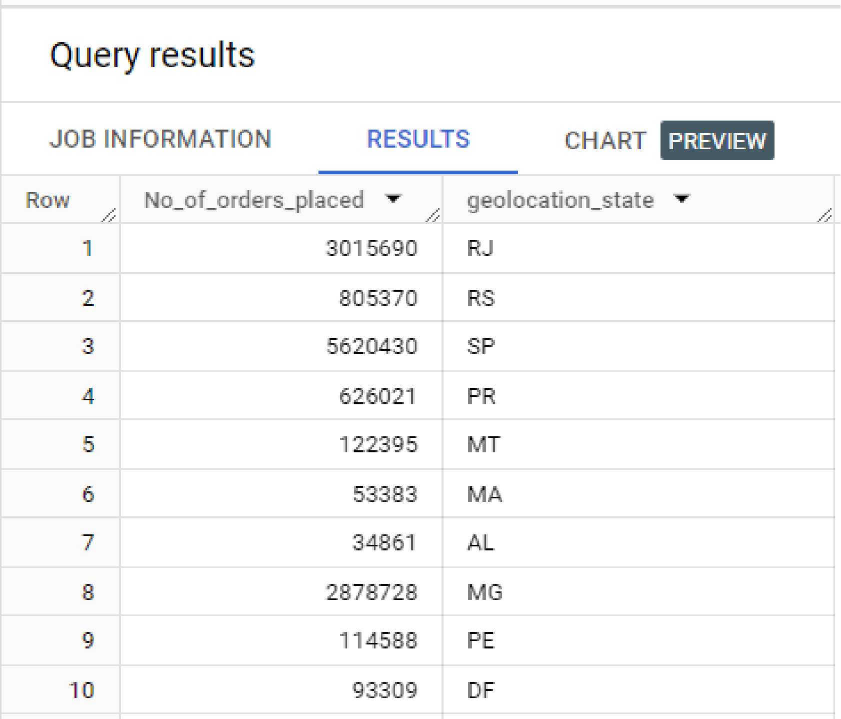
Insights: The query shows the number of orders placed in each state of Brazil in each month for the given time interval. The state RS could be seen having the maximum number of orders placed throughout, followed by the states SC and SP. Some more states with high order volume are BA, ES, MG, PR and RJ.

Recommendations: Following are the recommendations based on the insights:

a. The storage capacity for the above states should be increased seeing the demand,
b. Logistics network should also be made stronger to ensure on time and quick deliveries

2. How are the customers distributed across all the states?
Query -

```
select count(o.order_id) as `No_of_orders_placed`,
       g.geolocation_state
from target.orders as o
inner join target.customers as c
on o.customer_id=c.customer_id
inner join target.geolocation as g
on c.customer_zip_code_prefix=g.geolocation_zip_code_prefix
group by g.geolocation_state
```



Row	No_of_orders_placed	geolocation_state
1	3015690	RJ
2	805370	RS
3	5620430	SP
4	626021	PR
5	122395	MT
6	53383	MA
7	34861	AL
8	2878728	MG
9	114588	PE
10	93309	DF

Insights: The query shows the volume of orders placed from each state. It can be seen clearly that the states RJ, SP, MG, BA, PR and ES are the states contributing more to the total volume of orders being placed

Recommendations: Following are the recommendations as per the insights:

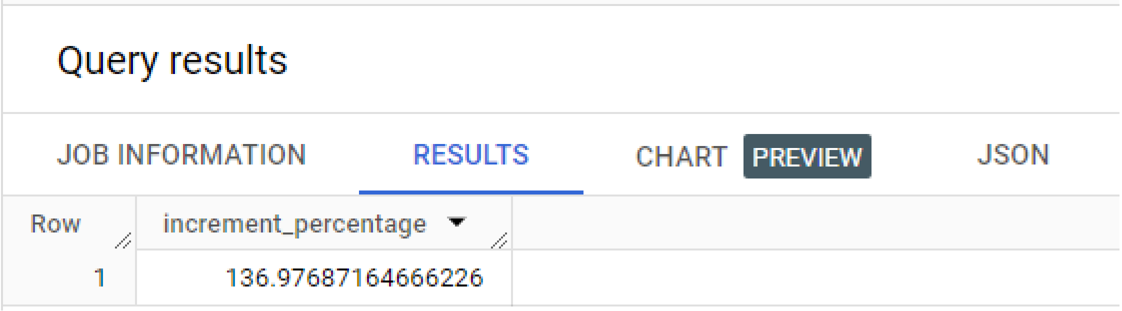
a. Region specific offers should be offered,
b. The availability of products should be ensured more for these states,
c. Some regional office can be setup for these regions to take care of the smooth procedure of delivery and more and also to provide better customer services.

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

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

Query -

```
with `cost_of_orders_2017` as (
       select sum(if(extract(month from
       o.order_purchase_timestamp)between 1 and 8,payment_value,0))
       as cost_of_orders
       from target.orders as o
       inner join target.payments as p on o.order_id=p.order_id
       where extract(year from o.order_purchase_timestamp)=2017),
       `cost_of_orders_2018` as (
       select sum(if(extract(month from
       o.order_purchase_timestamp)between 1 and  8,payment_value,0))
       as cost_of_orders
       from target.orders as o
       inner join target.payments as p on o.order_id=p.order_id
       where extract(year from o.order_purchase_timestamp)=2018)
select ((select cost_of_orders from `cost_of_orders_2018`)-(select
cost_of_orders from `cost_of_orders_2017`))*100/(select
cost_of_orders from `cost_of_orders_2017`) as increment_percentage
```



Row	increment_percentage
1	136.97687164666226

Insights: The query shows a significant increase in the number of orders in the given time frame. The order volume increased by approximately 137% over the year which is a big number.

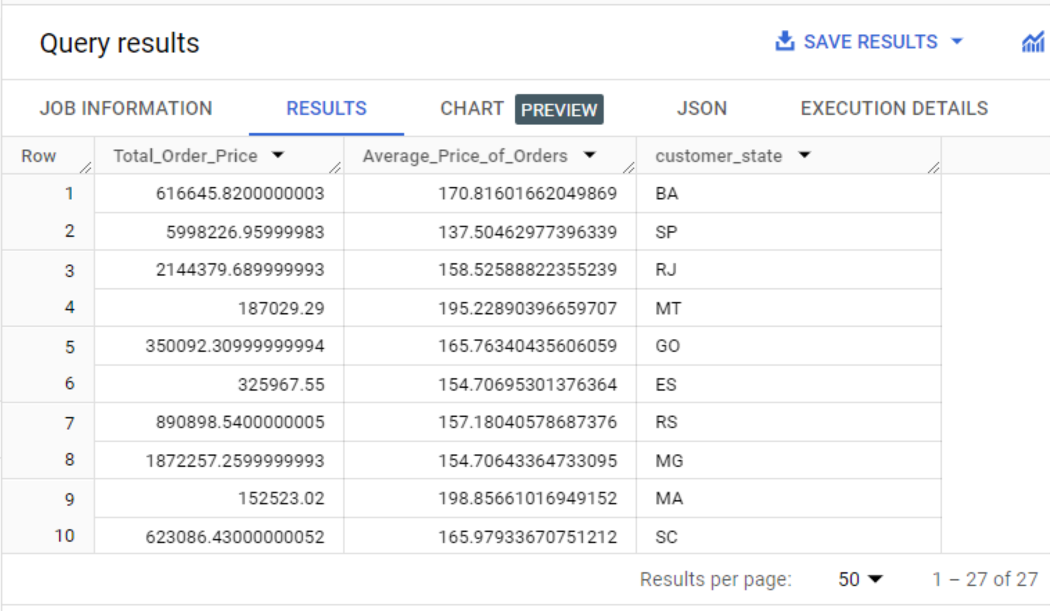
Recommendations: Following are the recommendations as per the insights:

a. The firm should now focus on increasing the company capacity to work,
b. The website should be upgraded to take more traffic and load,
c. More of support staffs should be hired for customer's convenience,
d. The product domain and range should be now expanded more,
e. The storage capacity and logistics infrastructure need to be strengthened.

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

Query -

```
select sum(p.payment_value) as `Total_Order_Price`,
       sum(p.payment_value)/count(o.order_id) as
       `Average_Price_of_Orders`,
       c.customer_state
from target.orders as o
inner join target.payments as p
on o.order_id=p.order_id
inner join target.customers as c
on o.customer_id=c.customer_id
group by c.customer_state
```



Row	Total_Order_Price	Average_Price_of_Orders	customer_state
1	616645.8200000003	170.81601662049869	BA
2	5998226.95999983	137.50462977396339	SP
3	2144379.689999993	158.52588822355239	RJ
4	187029.29	195.22890396659707	MT
5	350092.30999999994	165.76340435606059	GO
6	325967.55	154.70695301376364	ES
7	890898.5400000005	157.18040578687376	RS
8	1872257.2599999993	154.70643364733095	MG
9	152523.02	198.85661016949152	MA
10	623086.43000000052	165.97933670751212	SC

Results per page: 50 1 – 27 of 27

Insights: The query shows the total amount of orders placed and average of the same, for each state. It can be seen that the states like BA and SC contributes significantly to the amount.

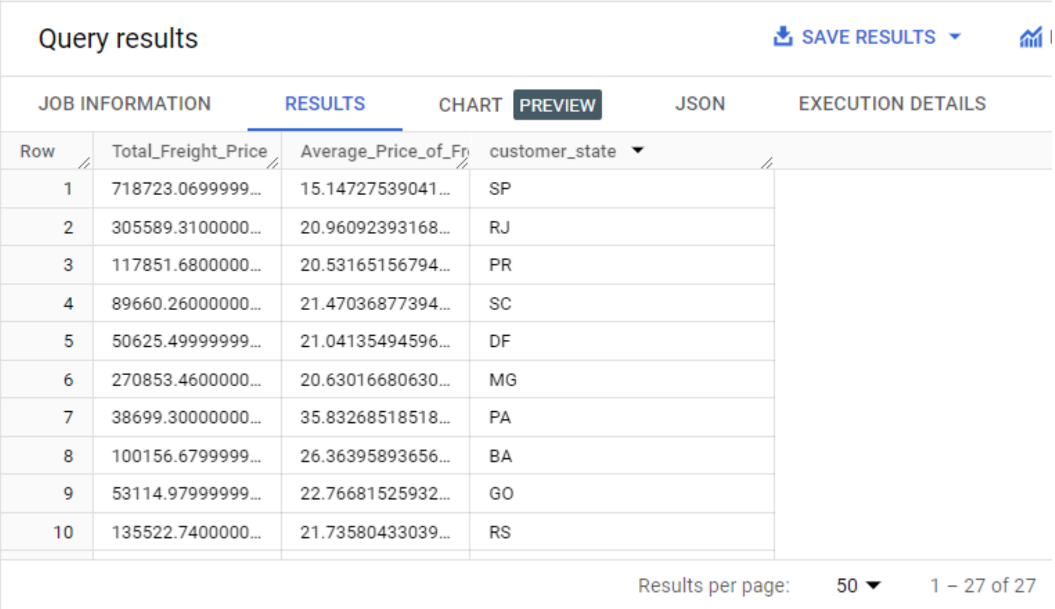
Recommendations: Following are the recommendations as per the insights:

a. Region specific offers should be offered,
b. The availability of products should be ensured more for these states,
c. Some regional office can be setup for these regions to take care of the smooth procedure of delivery and more and also to provide better customer services.

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

Query -

```
select sum(oi.freight_value) as `Total_Freight_Price`,
       sum(oi.freight_value)/count(oi.order_id) as
       `Average_Price_of_Freight`,
       c.customer_state
from target.order_items as oi
inner join target.orders as o
on oi.order_id=o.order_id
inner join target.customers as c
on o.customer_id=c.customer_id
group by c.customer_state
```



Row	Total_Freight_Price	Average_Price_of_Fr	customer_state
1	718723.0699999...	15.14727539041...	SP
2	305589.3100000...	20.96092393168...	RJ
3	117851.6800000...	20.53165156794...	PR
4	89660.26000000...	21.47036877394...	SC
5	50625.49999999...	21.04135494596...	DF
6	270853.4600000...	20.63016680630...	MG
7	38699.30000000...	35.83268518518...	PA
8	100156.6799999...	26.36395893656...	BA
9	53114.97999999...	22.76681525932...	GO
10	135522.7400000...	21.73580433039...	RS

Results per page: 50 1 – 27 of 27

Insights: The query shows the total amount of freight cost and average of the same, for each state. It can be seen that the state of SP, RJ, PR etc has high freight cost.

Recommendations: Following are the recommendations as per the insights:

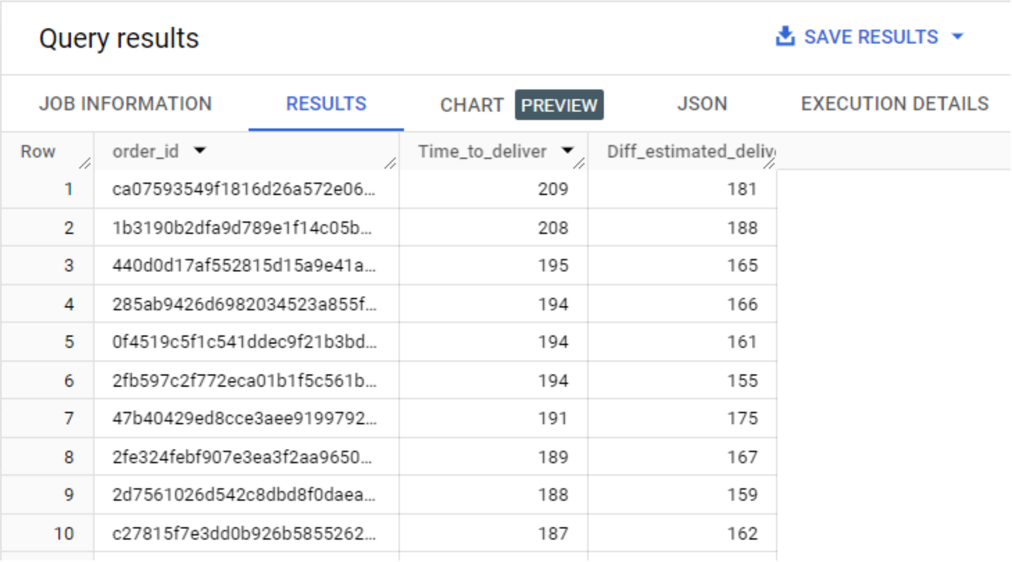
a. The freight delivery system needs to be optimised for the states with high freight cost to reduce the amount being spent,
b. The firm can look forward to setup its own delivery systems in these states by replacing the third-party facilities, wherever so.
c. More warehouses should be setup at these places to make the availability of products locally.

Analysis based on sales, freight and delivery time.

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.

Query -

```
select order_id,
       timestamp_diff(order_delivered_customer_date,order_purchase_t
       imestamp,day) as `Time_to_deliver`,
       abs(timestamp_diff(order_delivered_customer_date,order_estima
       ted_delivery_date,day)) as `Diff_estimated_delivery`
from `target.orders`
where order_delivered_customer_date is not null
order by `Time_to_deliver` desc,`Diff_estimated_delivery` desc
```



Row	order_id	Time_to_deliver	Diff_estimated_deliv
1	ca07593549f1816d26a572e06...	209	181
2	1b3190b2dfa9d789e1f14c05b...	208	188
3	440d0d17af552815d15a9e41a...	195	165
4	285ab9426d6982034523a855f...	194	166
5	0f4519c5f1c541ddec9f21b3bd...	194	161
6	2fb597c2f772eca01b1f5c561b...	194	155
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324febf907e3ea3f2aa9650...	189	167
9	2d7561026d542c8dbd8f0daea...	188	159
10	c27815f7e3dd0b926b5855262...	187	162

Insights: The query shows the estimated time to deliver an order and the actual time taken for its delivery. It shows that the orders aren't getting delivered on time.

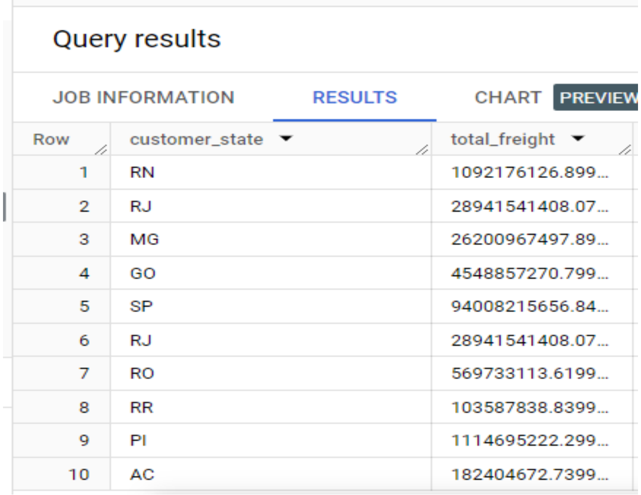
Recommendations: Following are the recommendations as per the insights:

a. The delivery infrastructure needs to be strengthened,
b. More delivery staffs are needed to be hired,
c. Product availability on time should be ensured,
d. More of regional warehouses should be setup for quick deliveries.

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

Query -

```
with c as(select c.customer_state,
       o.order_id,
       sum(oi.freight_value) as `total_freight_value`,
       avg(oi.freight_value) as `avg_freight_value`
       from `target.customers` as c
       inner join `target.orders` as o on c.customer_id=o.customer_id
       inner join `target.order_items` as oi on o.order_id=o.order_id
       group by 1,2),
     d as(select customer_state,
       sum(total_freight_value) as `total_freight` ,
       avg(avg_freight_value) as `avg_freight`
       from c
       group by 1
       order by 3 desc
       limit 5),
     e as(select customer_state,
       sum(total_freight_value) as `total_freight` ,
       avg(avg_freight_value) as `avg_freight`
       from c
       group by 1
       order by 3
       limit 5)
select *
from d
union all
select *
from e
order by `avg_freight`
```



Query results

Row	customer_state	total_freight
1	RN	1092176126.899...
2	RJ	28941541408.07...
3	MG	26200967497.89...
4	GO	4548857270.799...
5	SP	94008215656.84...
6	RJ	28941541408.07...
7	RO	569733113.6199...
8	RR	103587838.8399...
9	PI	1114695222.299...
10	AC	182404672.7399...

Insights: The first five rows show the top five states whereas, the next five shows the bottom five states according to the freight value.

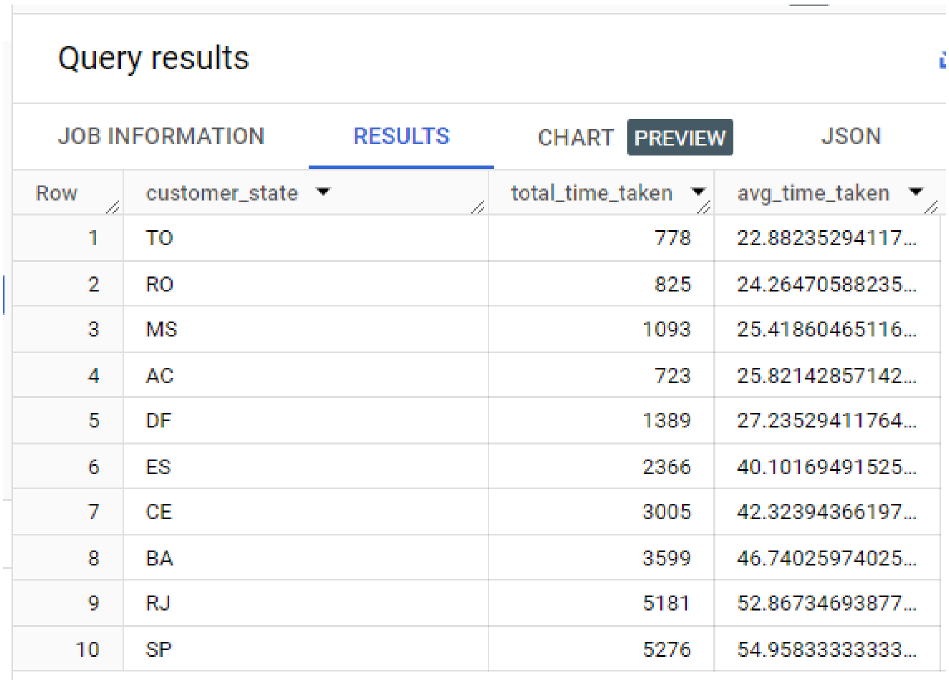
Recommendations: Following are the recommendations as per the insights:

a. The freight delivery system needs to be optimised for the states with high freight cost to reduce the amount being spent.

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

Query -

```
with c as(select c.customer_state,
     timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)
     as`time_taken_to_deliver`,
     avg(timestamp_diff
     (o.order_delivered_customer_date,o.order_purchase_timestamp,day))
     as `avg_time_taken_to_deliver`
       from `target.customers`as c
       inner join `target.orders` as o on c.customer_id=o.customer_id
       where o.order_status is not null
       group by 1,2),
     d as(select customer_state,
       sum(time_taken_to_deliver) as `total_time_taken`,
       avg(avg_time_taken_to_deliver) as `avg_time_taken`
       from c
       group by 1
       order by 3
       limit 5),
     e as(select customer_state,
       sum(time_taken_to_deliver) as `total_time_taken`,
       avg(avg_time_taken_to_deliver) as `avg_time_taken`
       from c
       group by 1
       order by 3 desc
       limit 5)
select *
from d
union all
select *
from e
order by `avg_time_taken`
```



Query results

Row	customer_state	total_time_taken	avg_time_taken
1	TO	778	22.88235294117...
2	RO	825	24.26470588235...
3	MS	1093	25.41860465116...
4	AC	723	25.82142857142...
5	DF	1389	27.23529411764...
6	ES	2366	40.10169491525...
7	CE	3005	42.32394366197...
8	BA	3599	46.74025974025...
9	RJ	5181	52.86734693877...
10	SP	5276	54.95833333333...

Insights: The first five rows show the top five states whereas, the next five shows the bottom five states according to average time taken for delivering products

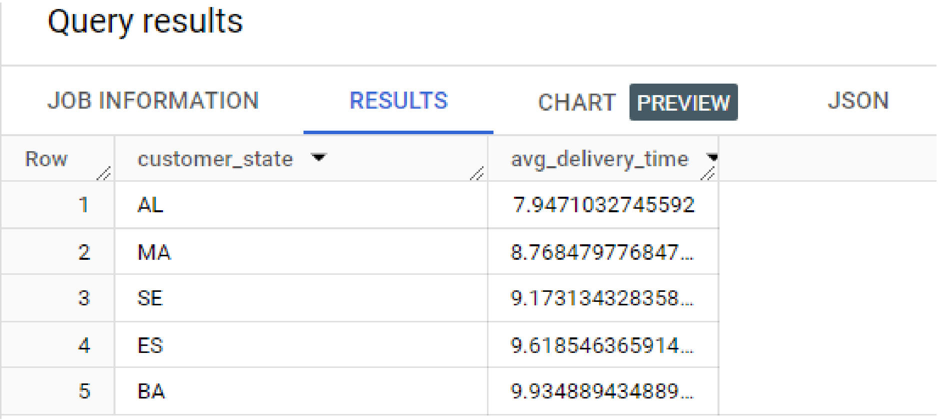
Recommendations: Following are the recommendations as per the insights:

a. The freight delivery system needs to be optimised to reduce the delivery time.

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Query -

```
with c as(select c.customer_state,
   avg(date_diff(
   o.order_estimated_delivery_date,o.order_deliver
   ed_customer_date,day)) as
       `avg_delivery_time`
       from `target.customers`as c
       inner join `target.orders` as o on
   c.customer_id=o.customer_id
       where o.order_delivered_customer_date is not null
       group by 1)
select *
from c
order by avg_delivery_time
limit 5
```



Row	customer_state	avg_delivery_time
1	AL	7.9471032745592
2	MA	8.768479776847...
3	SE	9.173134328358...
4	ES	9.618546365914...
5	BA	9.934889434889...

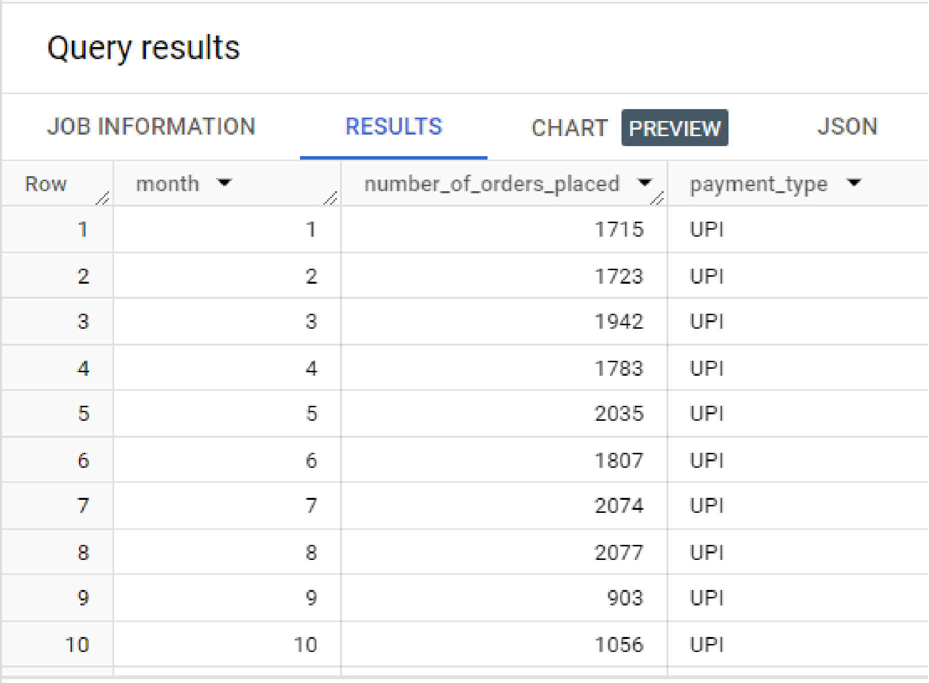
Insights: The query shows the top five states with the where the delivery is being done even before the estimated time of delivery and also, deliveries are being done quickest

Analysis based on the payments:

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

Query -

```
select extract(month from order_purchase_timestamp) as `month`,
       count(distinct o.order_id) as `number_of_orders_placed`,
       p.payment_type
from `target.orders` as o
join `target.payments` as p on o.order_id=p.order_id
group by `month`,p.payment_type
order by p.payment_type,`month`
```



Row	month	number_of_orders_placed	payment_type
1	1	1715	UPI
2	2	1723	UPI
3	3	1942	UPI
4	4	1783	UPI
5	5	2035	UPI
6	6	1807	UPI
7	7	2074	UPI
8	8	2077	UPI
9	9	903	UPI
10	10	1056	UPI

Insights: The query shows the numbers of orders placed per month as per the mode of payment used to place order. The above table shows the number of orders placed in the first ten months. The complete table shows that UPI mode was used for maximum of the payments but the count of payments done by credit card, debit card and vouchers are almost equal.

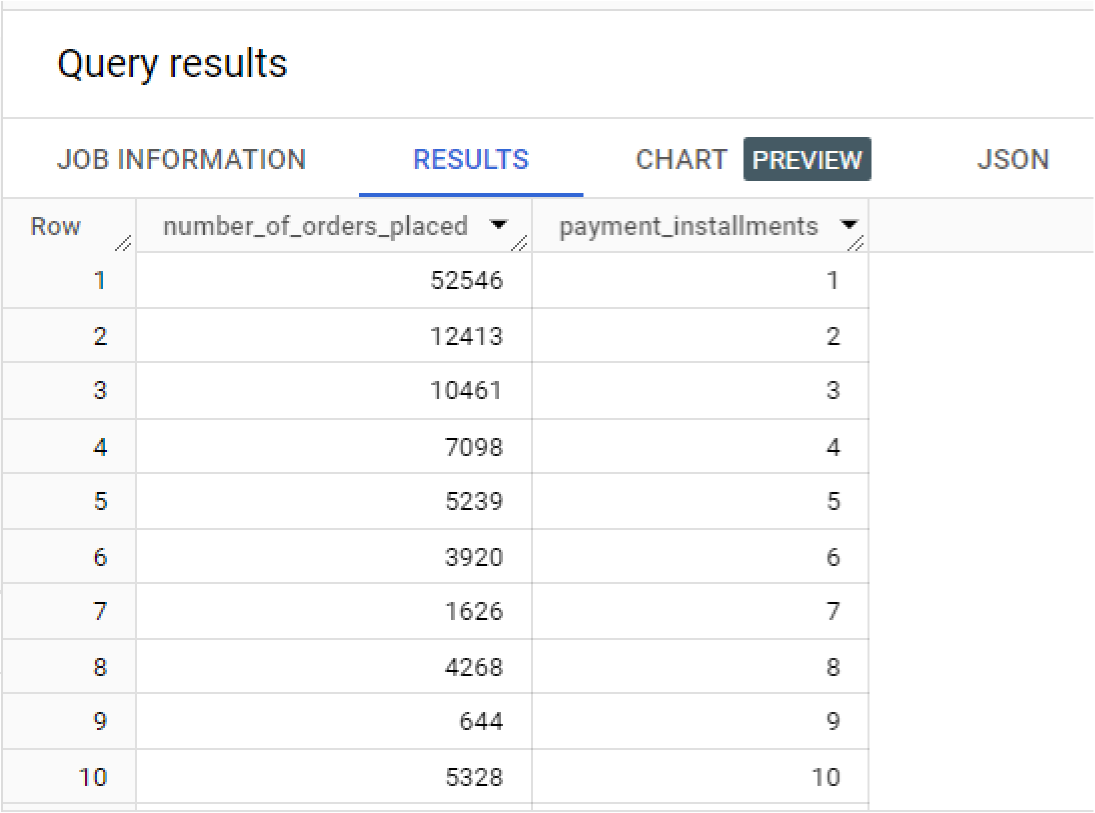
Recommendations: Following are the recommendations as per the insights:

a. Some special offers and cashback points can be introduced for the UPI payment mode,
b. Cashback point offers should be introduced for all these modes to promote more of online transactions.

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

Query -

```
select count(order_id) as number_of_orders_placed,
       payment_installments
from `target.payments`
where payment_installments>=1
group by payment_installments
```



Row	number_of_orders_placed	payment_installments
1	52546	1
2	12413	2
3	10461	3
4	7098	4
5	5239	5
6	3920	6
7	1626	7
8	4268	8
9	644	9
10	5328	10

Insights: The query shows the number of orders according to the number of instalments paid for them. It can be seen that 52,546 orders, which is the highest in the column, have one instalment cleared while, there are significantly smaller number of orders with more of their instalments cleared.

Recommendations: Following are the recommendations as per the insights:

a. Certain offers, like reduced interest rate, should be introduced for on time payback of instalments,
b. More of discounts on interest can be offered for pre-closure of the loan.