TARGET – BUSINESS CASE

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

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

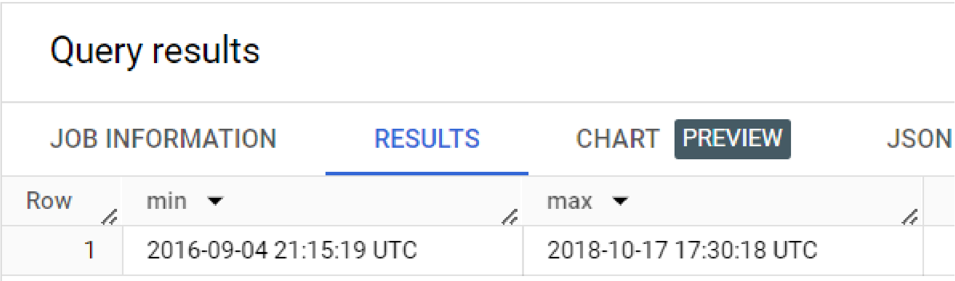
Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

Observation: The customers table contains two different datatypes, in which customer_id, customer_unique_id, customer_city, customer_state are STRING datatype and customer_zip_code_prefix is an INTEGER datatype.

Get the time range between which the orders were placed.

```
select min(order_purchase_timestamp) min,
       max(order_purchase_timestamp) max
 from `Target.orders`
```

Output:



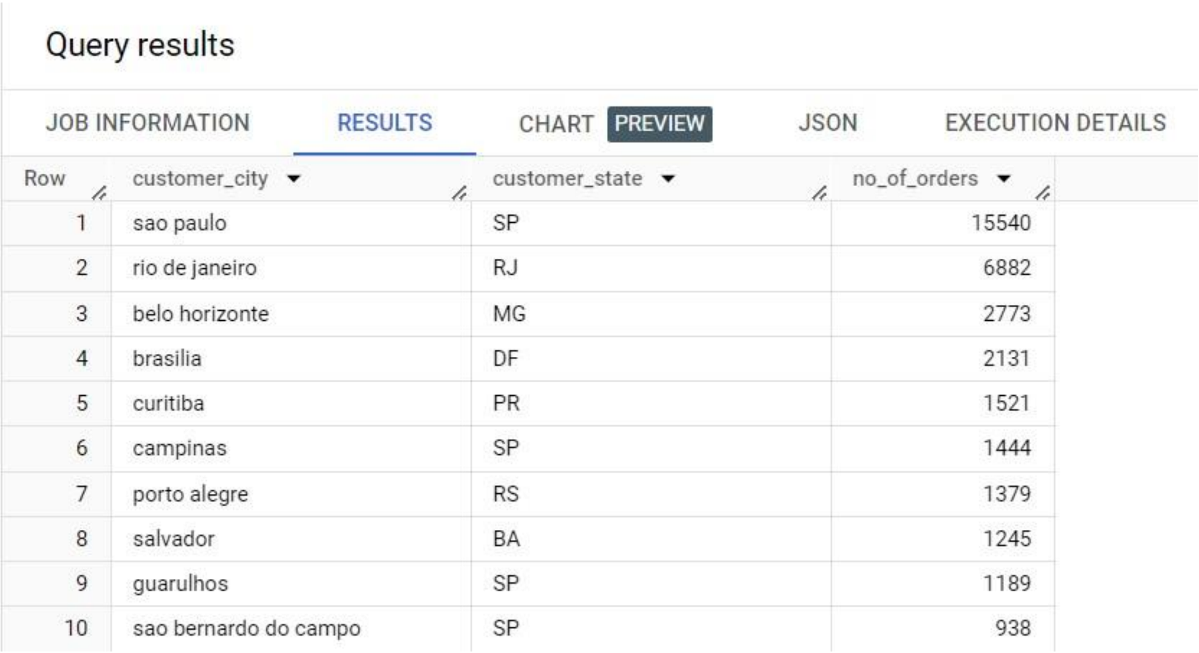
Row	min	max
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Observation: 2016, Sept 4th 21:15:19 UTC to 2018, Oct 17th 17:30:18 UTC was the time range between which orders were placed.

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

```
select distinct c.customer_city
                c.customer_state,
   count(c.customer_id) no_of_orders
   from `Target.orders` o
   join `Target.customers` c
   on o.customer_id = c.customer_id
   group by c.customer_city,
            c.customer_state
   order by no_of_orders desc
```

Output:



Query results

Row	customer_city	customer_state	no_of_orders
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521
6	campinas	SP	1444
7	porto alegre	RS	1379
8	salvador	BA	1245
9	guarulhos	SP	1189
10	sao bernardo do campo	SP	938

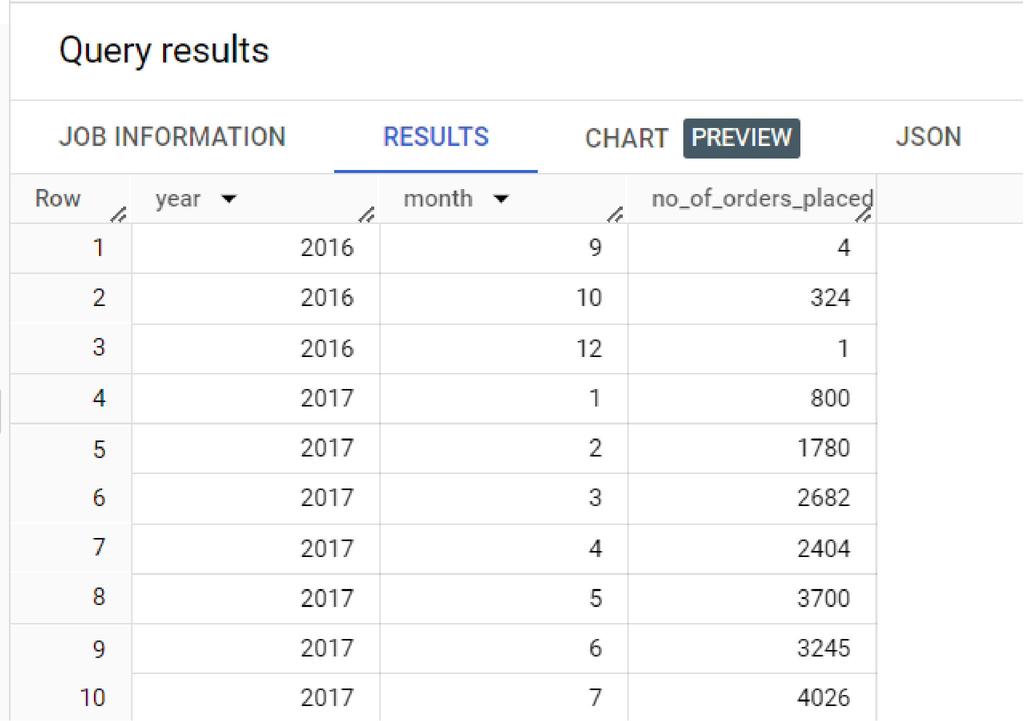
Observation: As for the above results a state named ‘SP’ in Brazil have high no. of orders from the customers when compared to the other states in the given time period.

In-depth Exploration:

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

```
select extract(year from order_purchase_timestamp) year,
       extract(month from order_purchase_timestamp) month,
       count(order_id) no_of_orders_placed
       from `Target.orders`
       group by year, month
  order by year, month
```

Output:



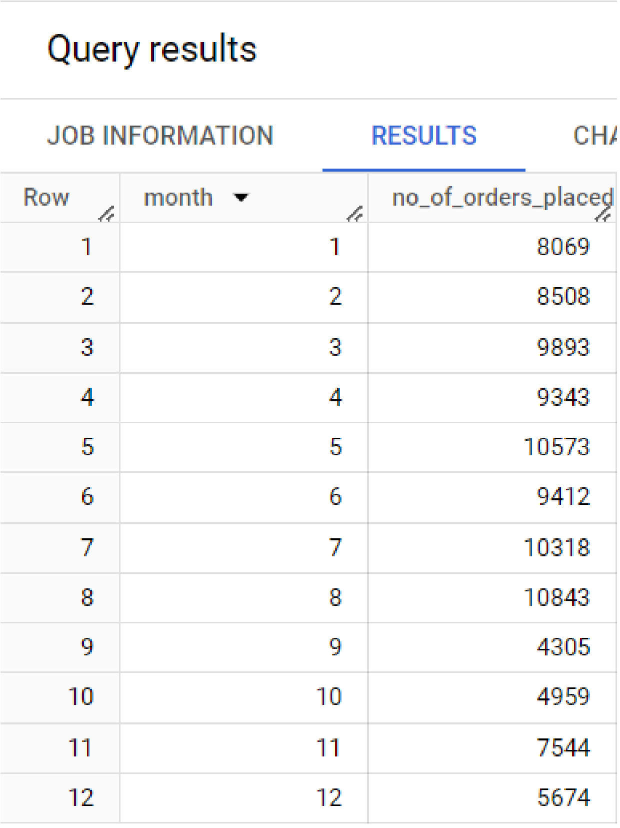
Row	year	month	no_of_orders_placed
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

Observation: Yes, there is a rapid growth in number of orders placed in the given time period and we can also see that monthly sales are increasing in numbers when compare to the past months and yearly wise as well.

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

```
select extract(month from order_purchase_timestamp) month,
            count(order_id) no_of_orders_placed
            from `Target.orders`
            group by month
            order by month
```

Output:


Query results

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
11	11	7544
12	12	5674

Observations:

Insights: According to the above results in the month of August no. of orders are at it peaks, also there is an rapid growth in the no. of orders from each month.

Recommendations: As we observe that since in the months of Nov, Jan and March no. of orders are high we can keep on increase the orders by

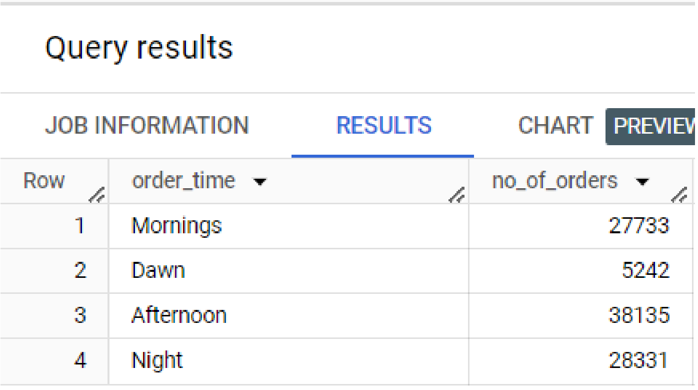
-by putting on offers
-by discounts
-by promoting as much as possible through different platforms
-by keeping the products in stock before the start of sales
-by having different types of varieties in products etc…

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

a. 0-6 hrs : Dawn
b. 7-12 hrs : Mornings
c. 13-18 hrs : Afternoon
d. 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 order_time,
            count(order_id) no_of_orders
            from `Target.orders`
            group by order_time
```

Output:



Query results

Row	order_time	no_of_orders
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

Observations: Count of orders from above tells us that most of the customers place there orders in the Afternoon mostly when they are free from there work and Night is the second next time that customers place there orders when they are at home then follows the next timings is Mornings and then at last the least time when customers place there orders is Dawn.

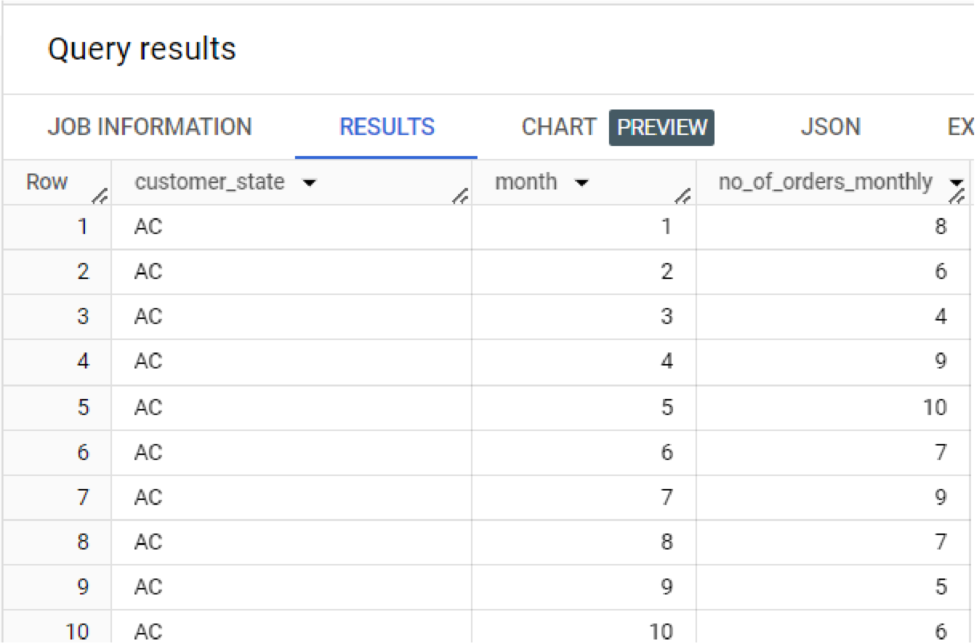
Evolution of E-commerce orders in the Brazil region:

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

```
select c.customer_state,
      extract(month from order_purchase_timestamp) month,
      count(o.order_id) no_of_orders_monthly
```

```
    from `Target.orders` o
    join `Target.customers` c
    on o.customer_id = c.customer_id
    group by c.customer_state,
             month
     order by c.customer_state,month
```

Output:



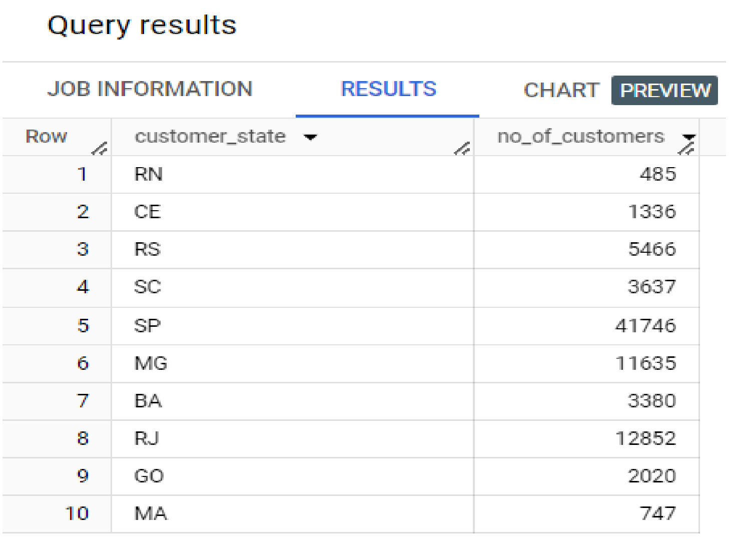
Row	customer_state	month	no_of_orders_monthly
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Observation: In the state ‘AL’ no. of orders placed are high when compared to the other states.

How are the customers distributed across all the states?

```
select customer_state,
       count(customer_id) no_of_customers
        from `Target.customers`
        group by customer_state
```

Output:



Query results

JOB INFORMATION RESULTS CHART PREVIEW

Row	customer_state	no_of_customers
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747

Observation: As per the number of customers in each state, SP state have the highest orders placed in this time period when compared to the other states in the Brazil.

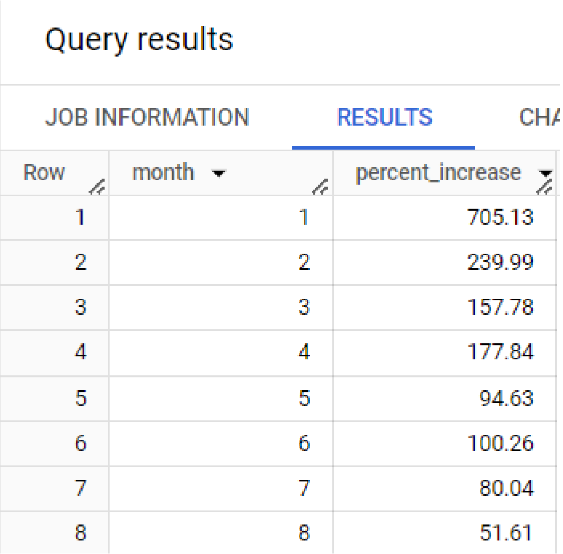
Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
You can use the "payment_value" column in the payments table to get the cost of orders.

```
select extract(month from o.order_purchase_timestamp) month,
       (
        (sum(case when extract(year from 
o.order_purchase_timestamp) = 2018 and
             extract(month from o.order_purchase_timestamp) 
between 1 and 8 then
             p.payment_value end)
             -
             sum(case when extract(year from 
o.order_purchase_timestamp) = 2017 and
             extract(month from o.order_purchase_timestamp) 
between 1 and 8 then
             p.payment_value end)
        )
        /
        sum(case when extract(year from 
o.order_purchase_timestamp) = 2017 and
```

```
             extract(month from o.order_purchase_timestamp)
between 1 and 8 then
             p.payment_value end)
       )* 100 percent_increase
       from `Target.orders` o
       join `Target.payments` p
       on o.order_id = p.order_id
       where extract(year from o.order_purchase_timestamp)
in (2017,2018) and
             extract(month from o.order_purchase_timestamp)
between 1 and 8
       group by 1
       order by 1
```

Output:



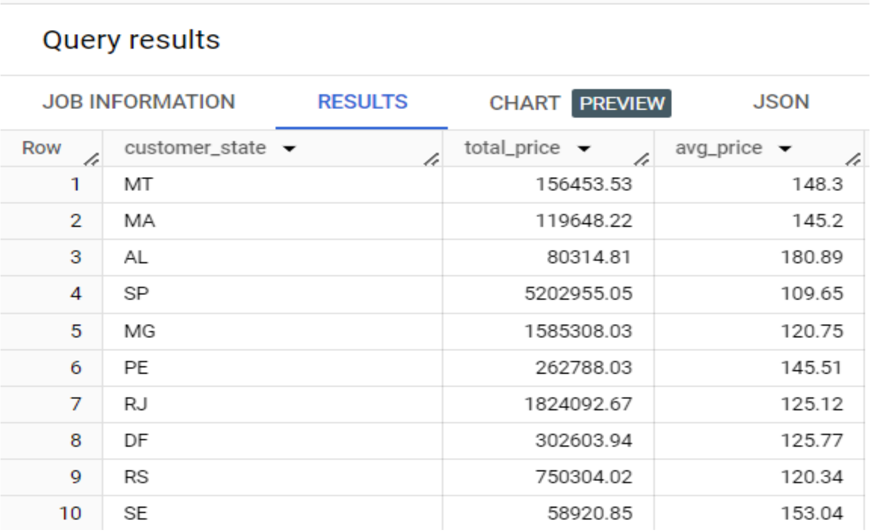
Row	month	percent_increase
1	1	705.13
2	2	239.99
3	3	157.78
4	4	177.84
5	5	94.63
6	6	100.26
7	7	80.04
8	8	51.61

Observation: From the years 2017 to 2018 the percent increase in cost of orders in the months of Jan to Aug, in which Jan month has been seen to be higher other than its following months.

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

```
select c.customer_state,
    round(sum(oi.price),2) total_price,
    round(avg(oi.price),2) avg_price
    from `Target.orders` o
    join `Target.customers` c
    on o.customer_id = c.customer_id
    join `Target.order_items` oi
    on o.order_id = oi.order_id
    group by c.customer_state
```

Output:



Query results

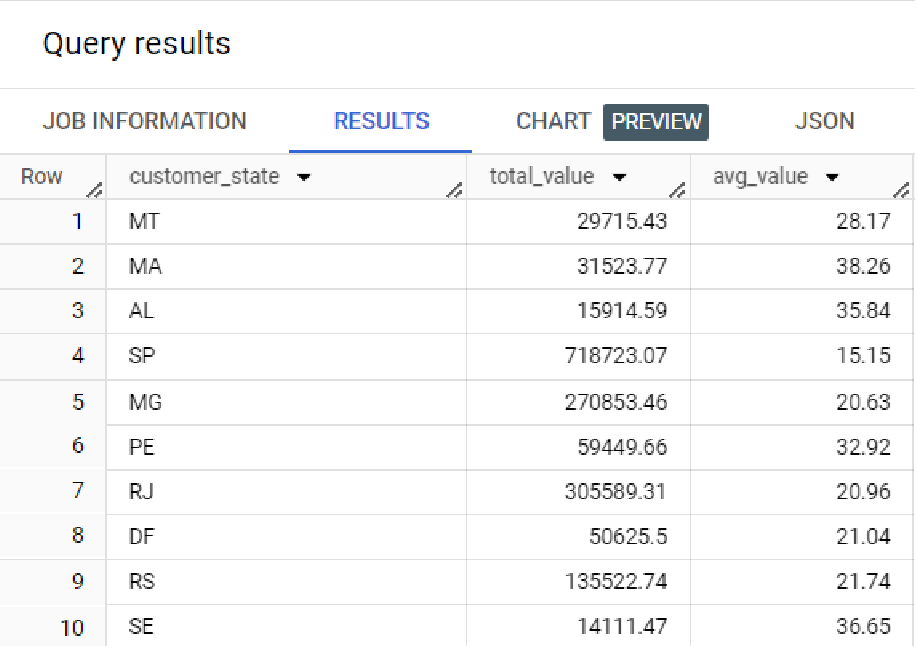
Row	customer_state	total_price	avg_price
1	MT	156453.53	148.3
2	MA	119648.22	145.2
3	AL	80314.81	180.89
4	SP	5202955.05	109.65
5	MG	1585308.03	120.75
6	PE	262788.03	145.51
7	RJ	1824092.67	125.12
8	DF	302603.94	125.77
9	RS	750304.02	120.34
10	SE	58920.85	153.04

Observation: By the above price analaysis of different states in Brazil we can draw a conclusion of customers avg and total amount variations in between each states.

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

```
select c.customer_state,
    round(sum(oi.freight_value),2) total_value,
    round(avg(oi.freight_value),2) avg_value
    from `Target.orders` o
    join `Target.customers` c
    on o.customer_id = c.customer_id
    join `Target.order_items` oi
    on o.order_id = oi.order_id
    group by c.customer_state
```

Output:



Row	customer_state	total_value	avg_value
1	MT	29715.43	28.17
2	MA	31523.77	38.26
3	AL	15914.59	35.84
4	SP	718723.07	15.15
5	MG	270853.46	20.63
6	PE	59449.66	32.92
7	RJ	305589.31	20.96
8	DF	50625.5	21.04
9	RS	135522.74	21.74
10	SE	14111.47	36.65

Observation: By the above freight analaysis of different states in Brazil we can draw a conclusion of customers avg and total amount variations in between each states.

Analysis based on sales, freight and delivery time.

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.

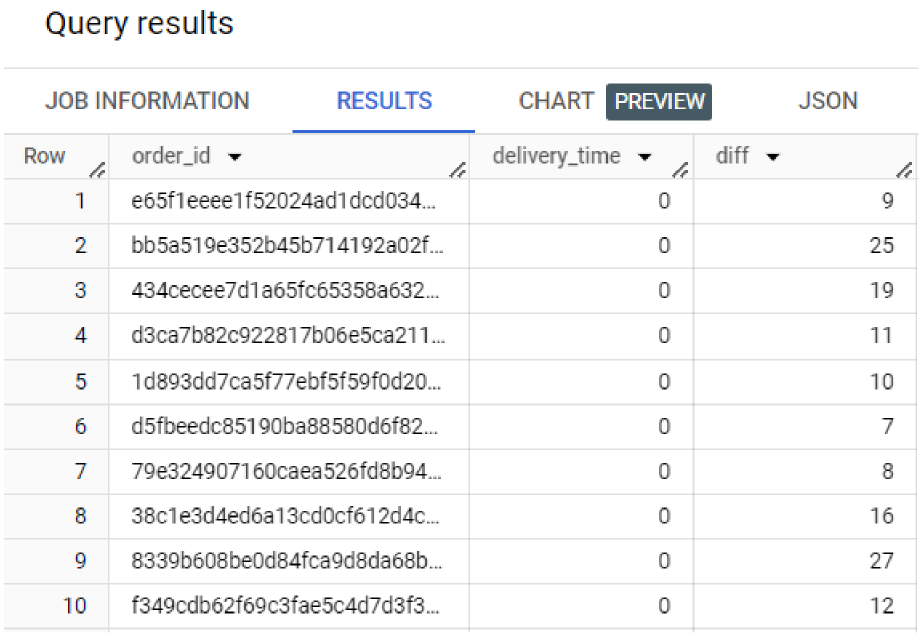
You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
- diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

```
select order_id,
        date_diff(order_delivered_customer_date,order_pu
rchase_timestamp,day) delivery_time,
        date_diff(order_estimated_delivery_date,order_deliver
ed_customer_date,day) diff
        from `Target.orders`
        where date_diff(order_delivered_customer_date,order_p
urchase_timestamp,day) is not null
```

```
          order by delivery_time
```

Output:



Query results

Row	order_id	delivery_time	diff
1	e65f1eeee1f52024ad1dcd034...	0	9
2	bb5a519e352b45b714192a02f...	0	25
3	434cecee7d1a65fc65358a632...	0	19
4	d3ca7b82c922817b06e5ca211...	0	11
5	1d893dd7ca5f77ebf5f59f0d20...	0	10
6	d5fbeedc85190ba88580d6f82...	0	7
7	79e324907160caea526fd8b94...	0	8
8	38c1e3d4ed6a13cd0cf612d4c...	0	16
9	8339b608be0d84fca9d8da68b...	0	27
10	f349cdb62f69c3fae5c4d7d3f3...	0	12

Observation: By the above difference in delivery of an product from its purchase date can be analysed and understand whether products to customers are delivered on or before the estimated time given to the customer.

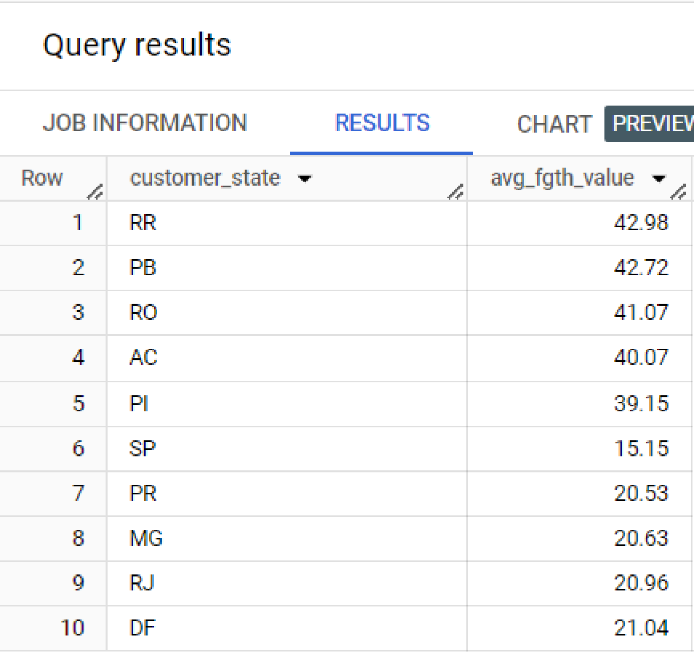
Find out the top 5 states with the highest & lowest average freight value.

```
(select c.customer_state,
       round(avg(freight_value),2) avg_fgth_value
       from `Target.orders` o
       join `Target.customers` c
       on o.customer_id = c.customer_id
       join `Target.order_items` oi
       on o.order_id = oi.order_id
       group by c.customer_state
       order by avg_fgth_value desc
       limit 5)

union all

(select c.customer_state,
       round(avg(freight_value),2) avg_fgth_value
       from `Target.orders` o
       join `Target.customers` c
       on o.customer_id = c.customer_id
       join `Target.order_items` oi
       on o.order_id = oi.order_id
       group by c.customer_state
       order by avg_fgth_value
       limit 5)
```

Output:



Row	customer_state	avg_fgth_value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15
6	SP	15.15
7	PR	20.53
8	MG	20.63
9	RJ	20.96
10	DF	21.04

Observation: By doing analysis of average of freight value in which top 5 states have highest average freight value are ‘RR,PB,RO,AC,PI’ and the lowest 5 states which have low average freight value are ‘SP,PR,MG,RJ,DF’.

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

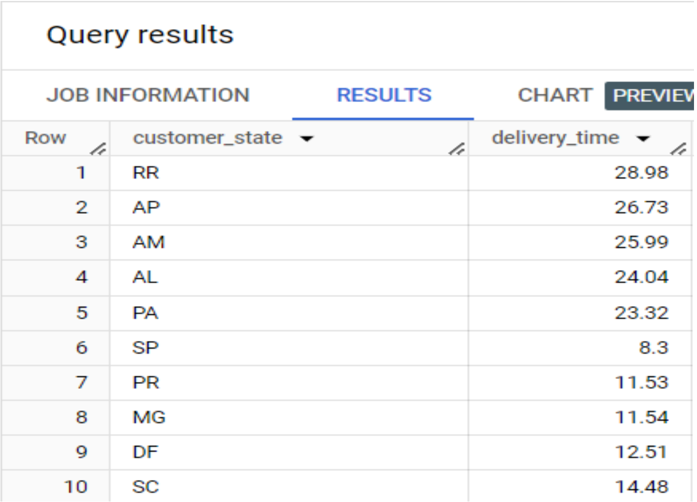
```
(select c.customer_state,

       round(avg(date_diff(order_delivered_customer_date,order_pu
r chase_timestamp,day)),2) delivery_time
       from `Target.orders` o
       join `Target.customers` c
       on o.customer_id = c.customer_id
       group by c.customer_state
       order by delivery_time desc
       limit 5)
```

```
union all

(select c.customer_state,
       round(avg(date_diff(order_delivered_customer_date,order_pu
rchase_timestamp,day)),2) delivery_time
       from `Target.orders` o
       join `Target.customers` c
       on o.customer_id = c.customer_id
       group by c.customer_state
       order by delivery_time
       limit 5)
```

Output:



Query results

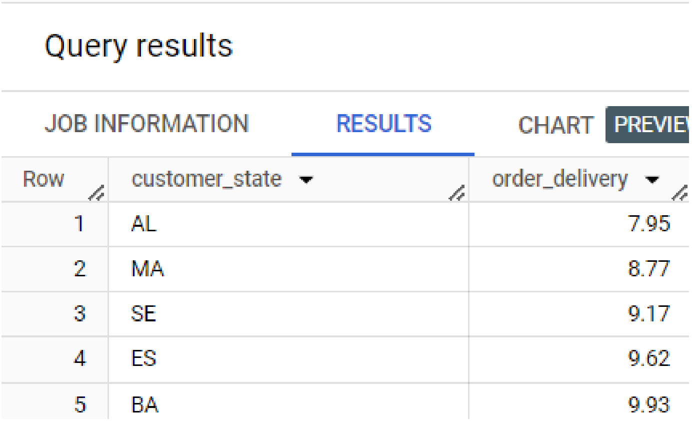
Row	customer_state	delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32
6	SP	8.3
7	PR	11.53
8	MG	11.54
9	DF	12.51
10	SC	14.48

Observation: By doing analysis of average of delivery time in which top 5 states have highest average delivery time are ‘RR,AP,AM,AL,PA’ and the lowest 5 states which have low average delivery time are ‘SP,PR,MG,DF,SC’.

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 c.customer_state,
       round(avg(date_diff(order_estimated_delivery_date,order_de
livered_customer_date,day)),2) order_delivery
       from `Target.orders` o
       join `Target.customers` c
       on o.customer_id = c.customer_id
       group by c.customer_state
       order by order_delivery
       limit 5
```

Order:



Query results

Row	customer_state	order_delivery
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

Observation: Fast delivery is one of the key business point, where the states which follow this key point and delivery the products before the estimated time of delivery are 'AL,MA,SE,ES,BA'.

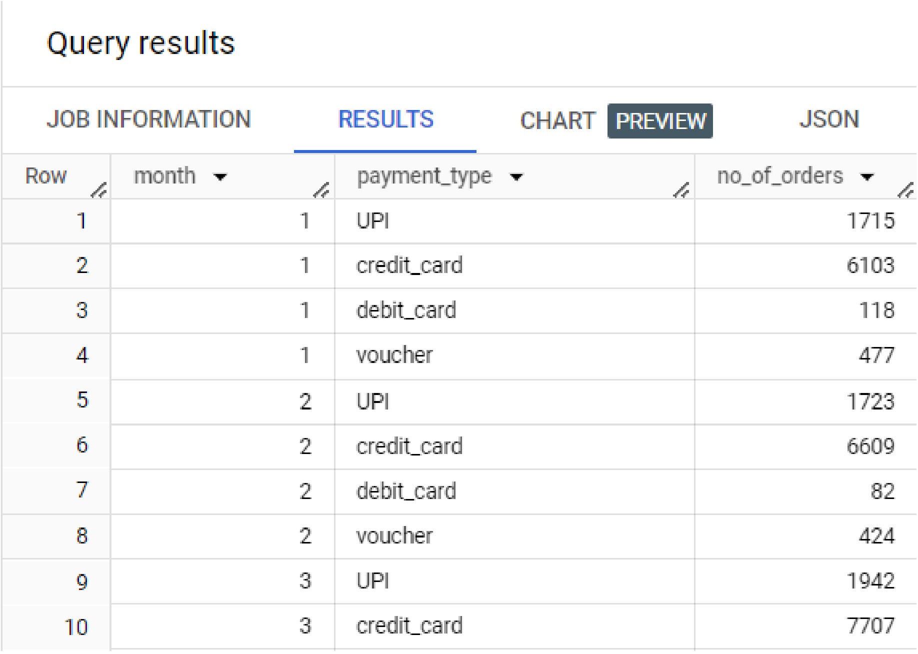
Analysis based on the payments:

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

```
select extract(month from order_purchase_timestamp) month,
       p.payment_type,
       count(o.order_id) no_of_orders
        from `Target.orders` o
        join `Target.payments` p
        on o.order_id = p.order_id
        group by month,
                 p.payment_type
        order by month,
                 p.payment_type
```

Output:

Output:



Query results

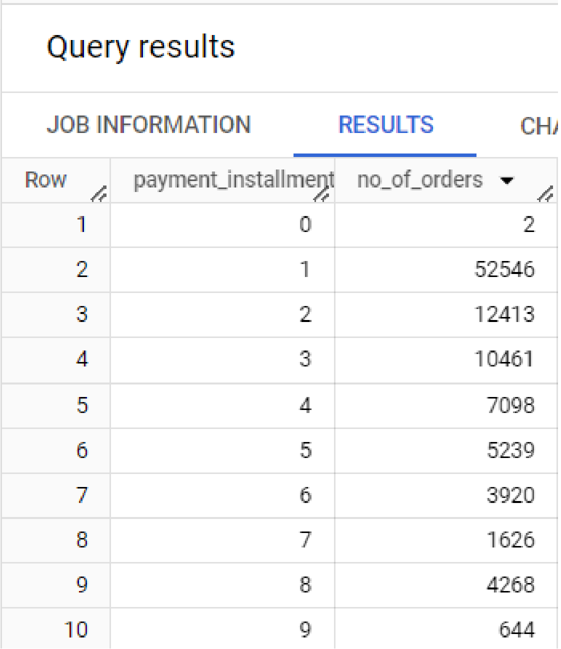
Row	month	payment_type	no_of_orders
1	1	UPI	1715
2	1	credit_card	6103
3	1	debit_card	118
4	1	voucher	477
5	2	UPI	1723
6	2	credit_card	6609
7	2	debit_card	82
8	2	voucher	424
9	3	UPI	1942
10	3	credit_card	7707

Observation: As there are different methods of payments by the above analysis we observe that customers in Brazil are tend to use credit card payment method most frequently and the next mostly used payment method is UPI method and then follows voucher and then debit_card.

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

```
select p.payment_installments,
       count(o.order_id) no_of_orders
       from `Target.orders` o
       join `Target.payments` p
       on o.order_id = p.order_id
       group by p.payment_installments
       order by p.payment_installments
```

Output:



Row	payment_installment	no_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Observation: The above results make us understand the customer preference upon payment installments. The count or no. of orders per each payment installment method explain us the most preferred payment installment method. By the above analysis we understand the customer preference and their payment methods.

...END...