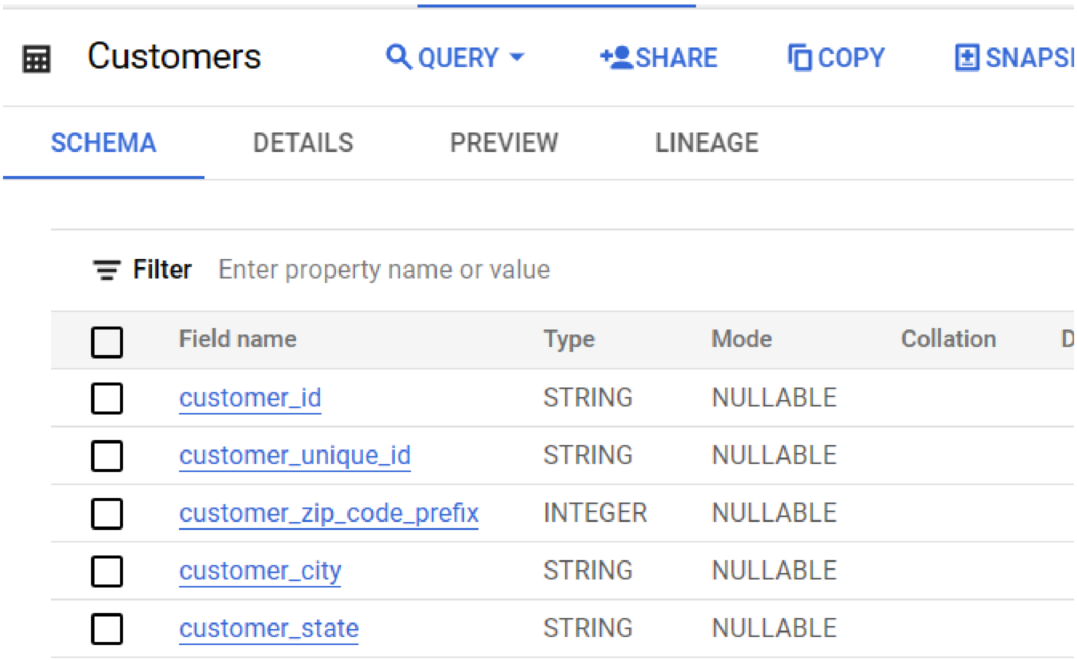
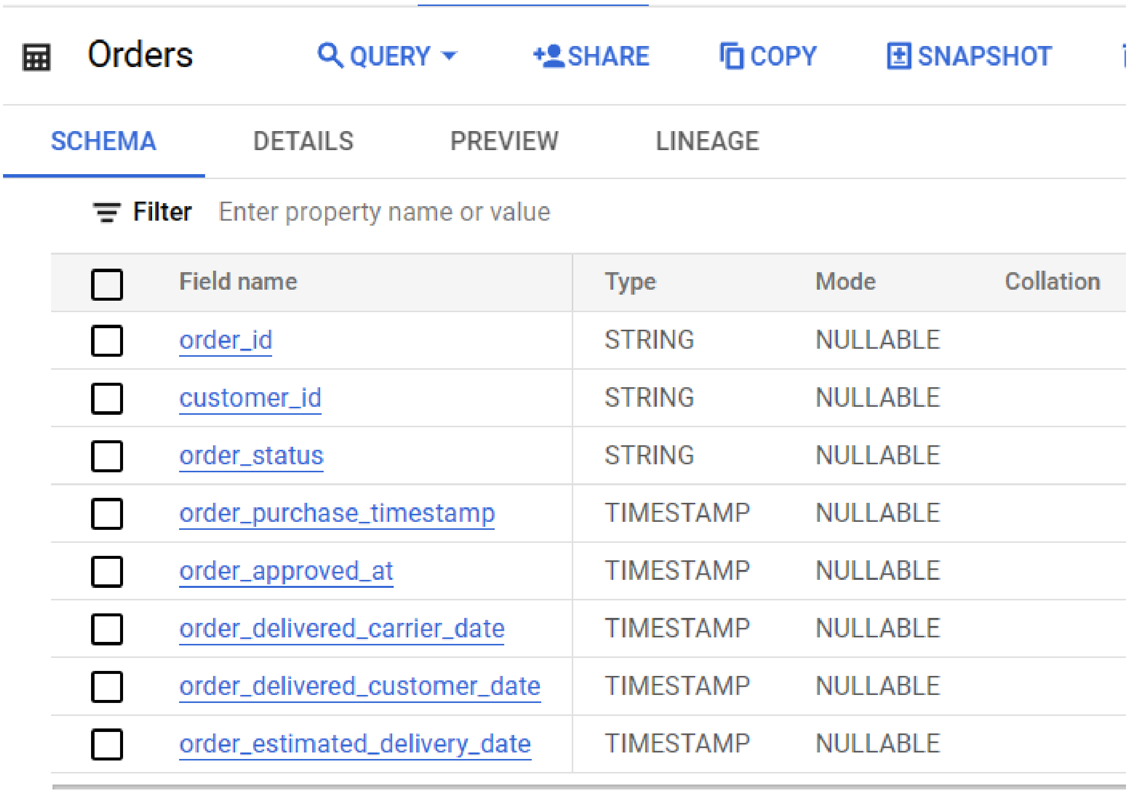
1. Data type of columns in a table



□	Field name	Type	Mode	Collation	D
□	customer_id	STRING	NULLABLE		
□	customer_unique_id	STRING	NULLABLE		
□	customer_zip_code_prefix	INTEGER	NULLABLE		
□	customer_city	STRING	NULLABLE		
□	customer_state	STRING	NULLABLE		

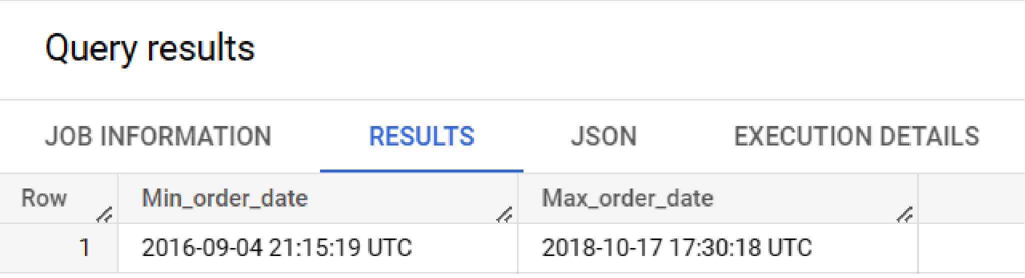


□	Field name	Type	Mode	Collation
□	order_id	STRING	NULLABLE	
□	customer_id	STRING	NULLABLE	
□	order_status	STRING	NULLABLE	
□	order_purchase_timestamp	TIMESTAMP	NULLABLE	
□	order_approved_at	TIMESTAMP	NULLABLE	
□	order_delivered_carrier_date	TIMESTAMP	NULLABLE	
□	order_delivered_customer_date	TIMESTAMP	NULLABLE	
□	order_estimated_delivery_date	TIMESTAMP	NULLABLE	

Data types of columns in Customers and Orders table.

2. Time period for which the data is given

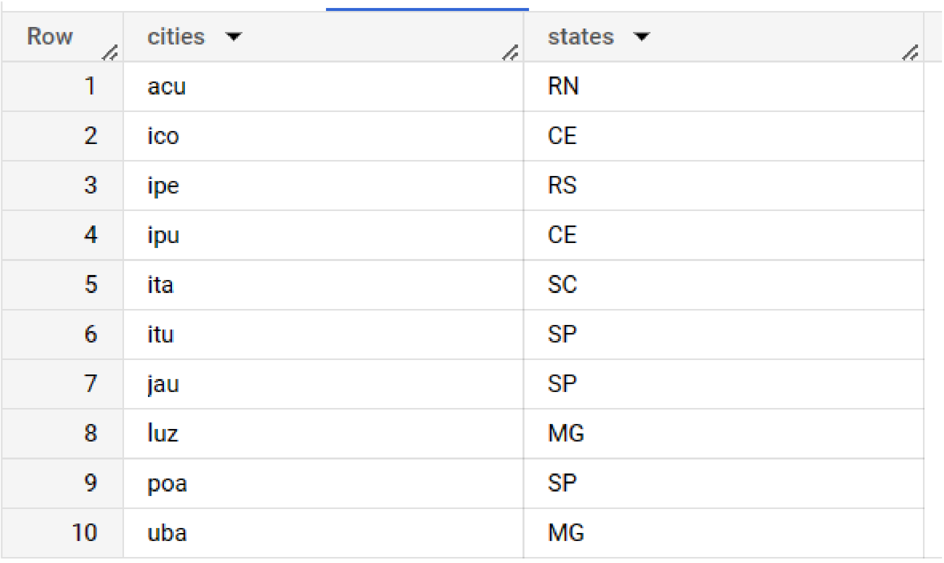
```
SELECT
        min(order_purchase_timestamp) as Min_order_date,
        max(order_purchase_timestamp) as Max_order_date
FROM
        `sqldemo-381616.Target_BusinessCase.Orders`;
```



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

3. Cities and States of customers ordered during the given period

```
    SELECT
            distinct customer_city as no_of_cities,
            customer_state as no_of_states
    FROM
            `sqldemo-381616.Target_BusinessCase.Customers` c
join
            `sqldemo-381616.Target_BusinessCase.Orders` o
    on
            c.customer_id = o.customer_id
    where
            o.order_purchase_timestamp
    between
            '2016-09-04 21:15:19' and  '2018-10-17 17:30:18'
    limit 10;
```



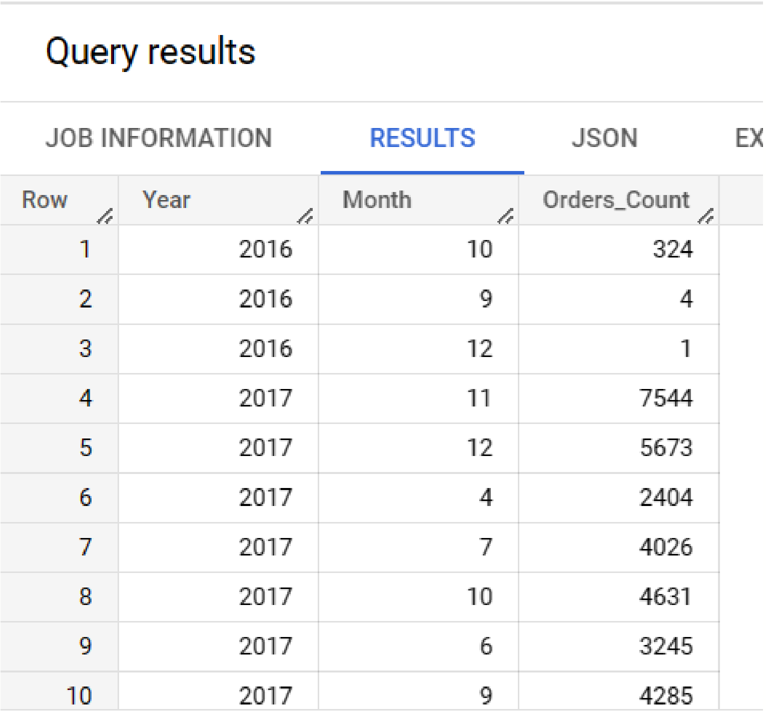
Row	cities	states
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC
6	itu	SP
7	jau	SP
8	luz	MG
9	poa	SP
10	uba	MG

City and state of customers who ordered in between given timestamp.

2.

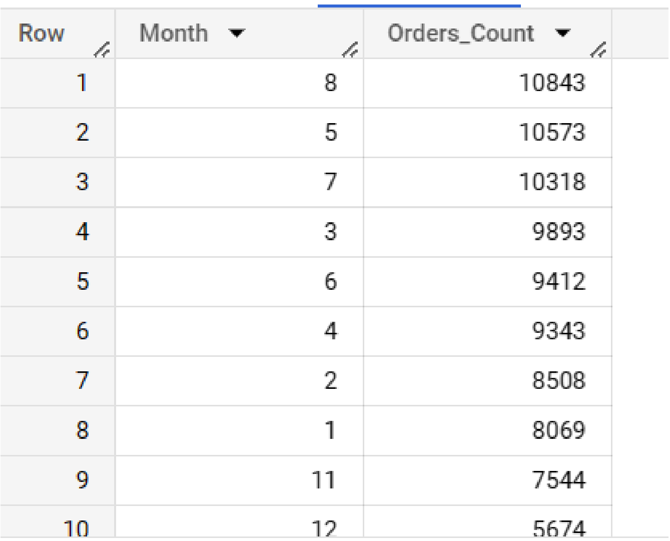
1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

```
SELECT
        extract  (year from order_purchase_timestamp ) as Year,
        extract  (month from order_purchase_timestamp ) as Month ,
        count(order_id) as  Orders_Count
FROM
        `sqldemo-381616.Target_BusinessCase.Orders`
group by
        Year,
        Month
order by
        Year,
        Month;
```



Row	Year	Month	Orders_Count
1	2016	10	324
2	2016	9	4
3	2016	12	1
4	2017	11	7544
5	2017	12	5673
6	2017	4	2404
7	2017	7	4026
8	2017	10	4631
9	2017	6	3245
10	2017	9	4285

❖ We can observe a growing trend in e-commerce in the Year 2017 while in the year 2018 there are fluctuations in trend.

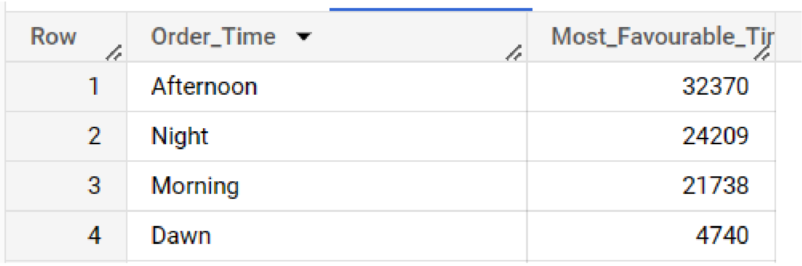


Row	Month	Orders_Count
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674

We can see that in the month of August, May and July number of orders are at peak.

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

```
SELECT
Order_Time,
Count(Order_Time) as Most_Favourable_Time
from
(Select
Case
when  extract(time from order_purchase_timestamp) between '7:00:00' and '12:00:00' then 'Morning'
when  extract(time from order_purchase_timestamp) between '13:00:00' and '18:00:00' then 'Afternoon'
when  extract(time from order_purchase_timestamp) between '19:00:00' and '23:00:00' then 'Night'
when  extract(time from order_purchase_timestamp) between '00:00:00' and '6:00:00' then 'Dawn'
END as
Order_Time
FROM
`sqldemo-381616.Target_BusinessCase.Customers` c join `sqldemo-381616.Target_BusinessCase.Orders` o on
c.customer_id = o.customer_id
)
group by
Order_Time
order by
Most_Favourable_Time desc;
```



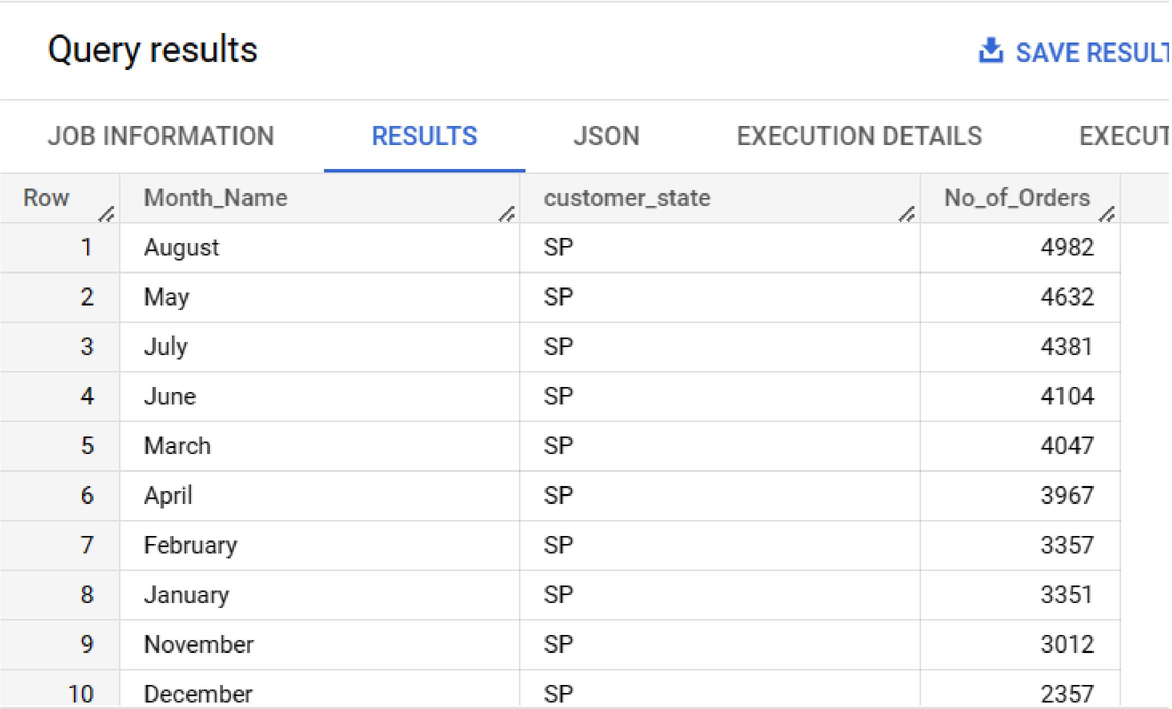
Row	Order_Time	Most_Favourable_Tir
1	Afternoon	32370
2	Night	24209
3	Morning	21738
4	Dawn	4740

- ❖ Afternoon time Brazilians tend to buy more.
- ❖ Very less people shop at late night, this is one area where Target can focus to improve sales during this time.

3.

1. Get month on month orders by states

```
SELECT

            FORMAT_DATETIME("%B",DATETIME (order_purchase_timestamp))

            as Month_Name,c.customer_state ,count(*)as No_of_Orders FROM `sqldemo-
            381616.Target_BusinessCase.Customers` c

left join

            `sqldemo-381616.Target_BusinessCase.Orders` o on c.customer_id = o.customer_id

 group by
            c.customer_state,Month_Name
 order by
            No_of_Orders desc
 LIMIT 1000;
```

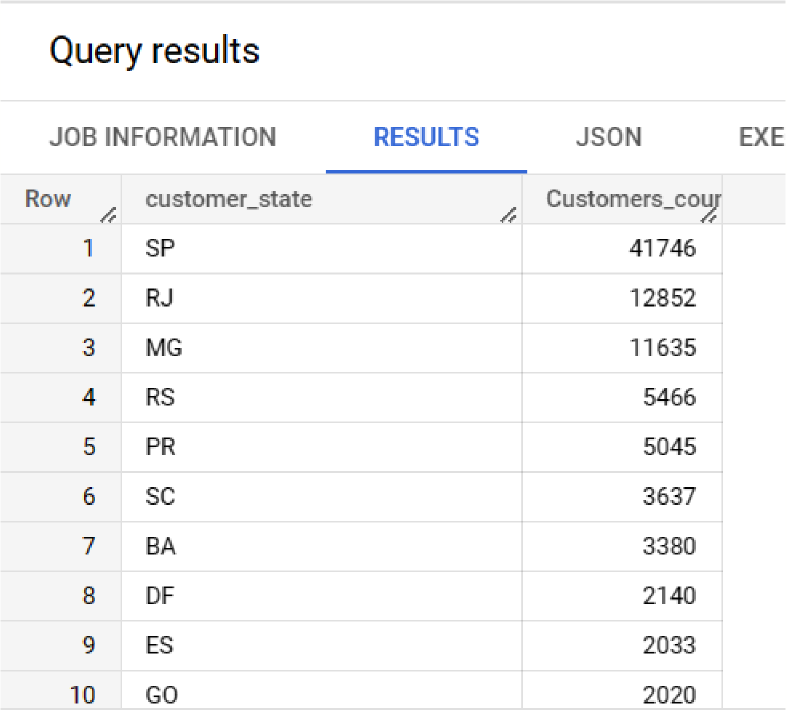


Row	Month_Name	customer_state	No_of_Orders
1	August	SP	4982
2	May	SP	4632
3	July	SP	4381
4	June	SP	4104
5	March	SP	4047
6	April	SP	3967
7	February	SP	3357
8	January	SP	3351
9	November	SP	3012
10	December	SP	2357

- State SP has most number of orders as to other states and AC, AP & RR have least no of orders.

2. Distribution of customers across the states in Brazil

```
SELECT
        customer_state,
        count(customer_unique_id) as Customers_count
FROM
        `sqldemo-381616.Target_BusinessCase.Customers`
group by
        customer_state
order by
        Customers_count desc
LIMIT 1000;
```



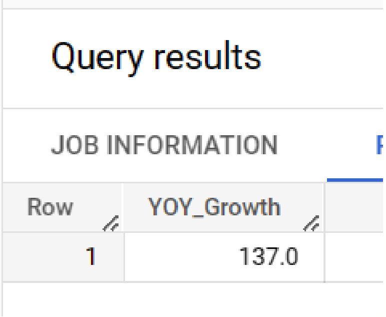
Row	customer_state	Customers_cour
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

- More than 2/3rd population lies in 3 states i.e. SP, RJ, MG
- almost 2/3rd of the customers is coming from 3 states. Target can focus on other states to attract more customers and boost sales

4.

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

```
SELECT round(((Sales_2 - Sales_1)/Sales_1)*100,0) as YOY_Growth from(

SELECT
sum(case when Year = 2017 and Month between 1 and 8 then payment_valueend) as Sales_1
,
sum(case when Year = 2018 and Month between 1 and 8 then payment_valueend) as Sales_2
from(

SELECT extract(Month from order_purchase_timestamp) as Month,extract(Year from order_pu
rchase_timestamp) as Year,
p.payment_value FROM `sqldemo-381616.Target_BusinessCase.Orders` o join `sqldemo-
381616.Target_BusinessCase.Payments` p on o.order_id = p.order_id));
```

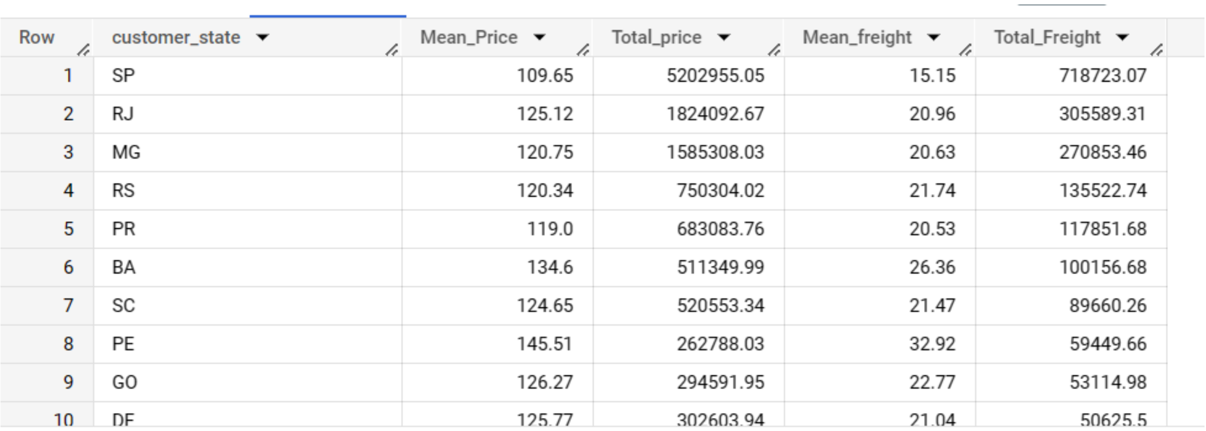


Row	YOY_Growth
1	137.0

- YOY Growth % is 137.

2. Mean & Sum of price and freight value by customer state

```
SELECT
      c.customer_state,
      round(avg(oi.price),2) as Mean_Price,
      round(sum(oi.price),2) as Total_price,
      round(avg(oi.freight_value),2) as Mean_freight,
      round(sum(oi.freight_value),2) as Total_Freight
FROM
      `sqldemo-381616.Target_BusinessCase.Customers` c
left join
      `sqldemo-381616.Target_BusinessCase.Orders` o
 on
      c.customer_id = o.customer_id join `sqldemo-381616.Target_BusinessCase.OrderItems` oi
 on
      o.order_id = oi.order_id

group by
      c.customer_state;
```



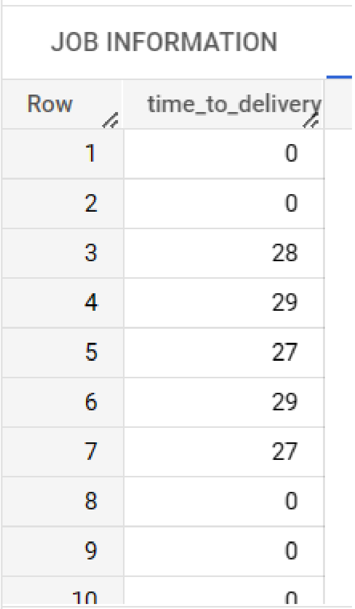
Row	customer_state	Mean_Price	Total_price	Mean_freight	Total_Freight
1	SP	109.65	5202955.05	15.15	718723.07
2	RJ	125.12	1824092.67	20.96	305589.31
3	MG	120.75	1585308.03	20.63	270853.46
4	RS	120.34	750304.02	21.74	135522.74
5	PR	119.0	683083.76	20.53	117851.68
6	BA	134.6	511349.99	26.36	100156.68
7	SC	124.65	520553.34	21.47	89660.26
8	PE	145.51	262788.03	32.92	59449.66
9	GO	126.27	294591.95	22.77	53114.98
10	DF	125.77	302603.94	21.04	50625.5

- SP, RJ & MG have highest freight value
- States like PR and RR have lowest freight value
- Difference between highest and lowest average freight value is very large
- There are states like RR, PR where freight is very high. these areas can be focused to cut operation cost related to freight

5.

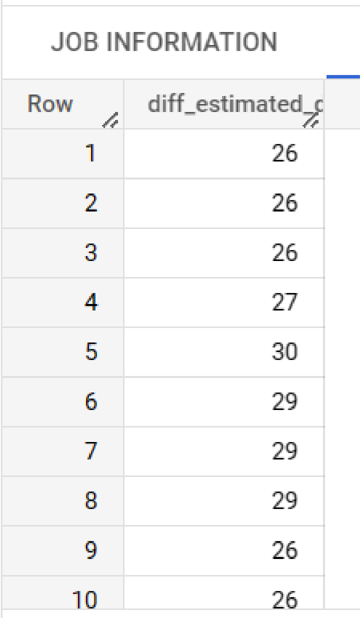
1. Calculate days between purchasing, delivering and estimated delivery

```
SELECT

            abs(extract(day from order_purchase_timestamp) - extract      (day
        from
                order_delivered_customer_date)) as days_to_delivery
        FROM
                `sqldemo-381616.Target_BusinessCase.Orders` ;

SELECT
        abs(extract(day from order_estimated_delivery_date) - extract (day from order_delivered_customer_date)) as diff_estimated_deliveryDays
FROM
        `sqldemo-381616.Target_BusinessCase.Orders` ;
```



JOB INFORMATION

Row	time_to_delivery
1	0
2	0
3	28
4	29
5	27
6	29
7	27
8	0
9	0
10	0



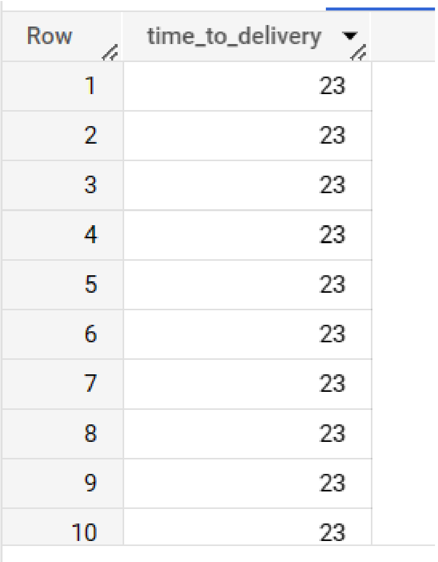
JOB INFORMATION

Row	diff_estimated_c
1	26
2	26
3	26
4	27
5	30
6	29
7	29
8	29
9	26
10	26

2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

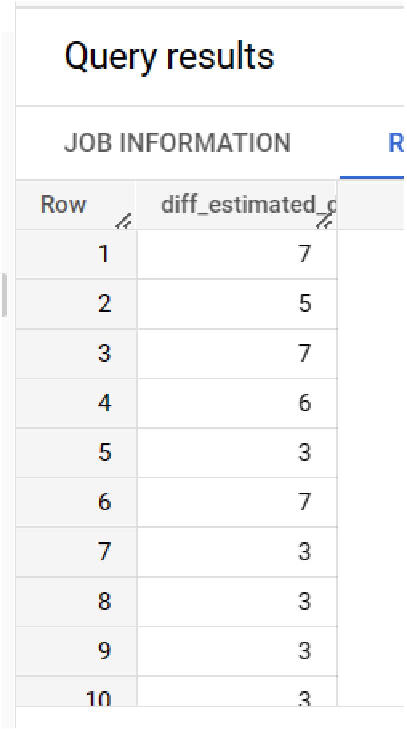
- time_to_delivery = order_purchase_timestamp - order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

```
SELECT
        abs(extract(hour from order_purchase_timestamp) - extract (hour from order_delivered_customer_date)) as time_to_delivery
FROM
        `sqldemo-381616.Target_BusinessCase.Orders` ;

SELECT
        abs(extract(hour from order_estimated_delivery_date) - extract (hour from order_delivered_customer_date)) as diff_estimated_delivery
FROM
        `sqldemo-381616.Target_BusinessCase.Orders` ;
```



Row	time_to_delivery
1	23
2	23
3	23
4	23
5	23
6	23
7	23
8	23
9	23
10	23

- Highest avg time to delivery is 28 days is in state RR
- avg difference of estimated vs delivered date ranges from 8-20 days. The variance can be improved to give smoother experience to customers
- Highest Avg time to deliver a product is 28 days which is very high. This can be worked upon to cut delivery time to make customers more satisfied.



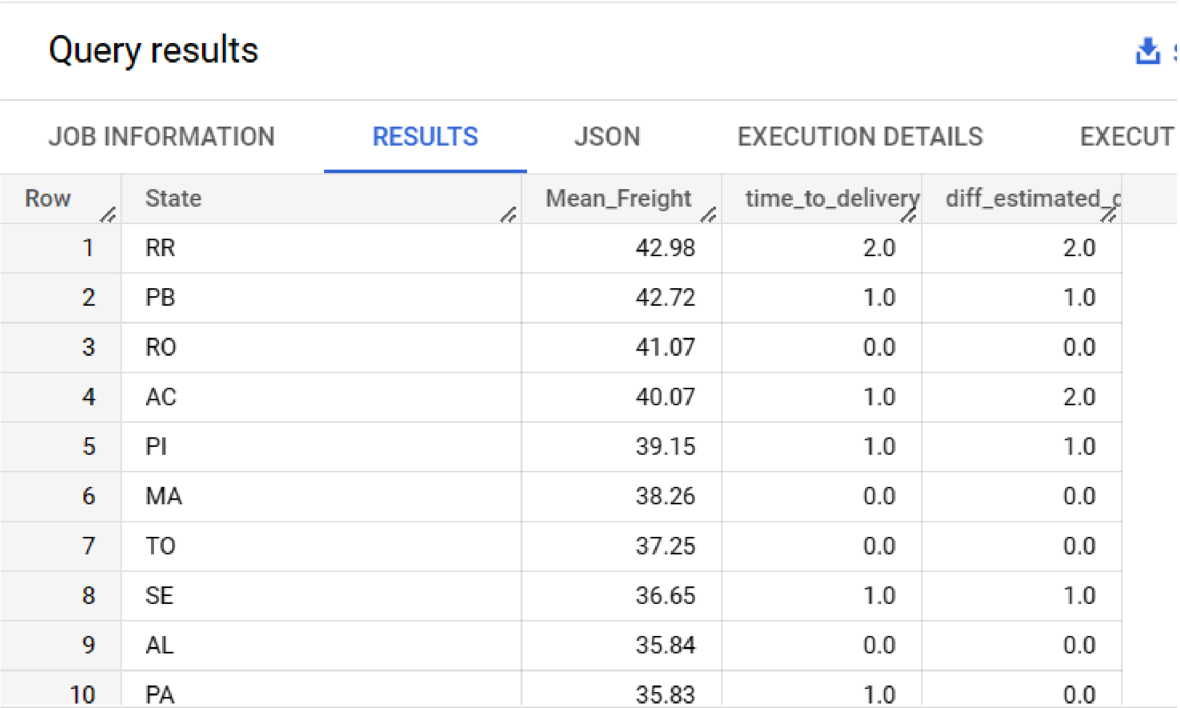
Query results

Row	diff_estimated_c
1	7
2	5
3	7
4	6
5	3
6	7
7	3
8	3
9	3
10	3

- ❖ SP has lowest avg time to delivery which is 8 days
- ❖ Avg difference of delivery vs estimated dates differ in the range of 8-20 days

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
            SELECT
              c.customer_state as State,
              Round(avg(oi.freight_value),2) as Mean_Freight,
              Round(abs(avg(extract(hour from o.order_purchase_timestamp)-
              extract(hour from o.order_delivered_customer_date))),0) as time_to_delivery
              ,
              Round(abs(avg(extract(day from o.order_estimated_delivery_date)-
              extract(day from o.order_delivered_customer_date))),0) as diff_estimated_de
              livery
        FROM
              `sqldemo-381616.Target_BusinessCase.Customers` c
        join
              `sqldemo-381616.Target_BusinessCase.Orders` o
        on
              c.customer_id = o.customer_id join `sqldemo-
              381616.Target_BusinessCase.OrderItems` oi
        on
              o.order_id = oi.order_id
        group by
              c.customer_state
        order by
              Mean_Freight desc;
```

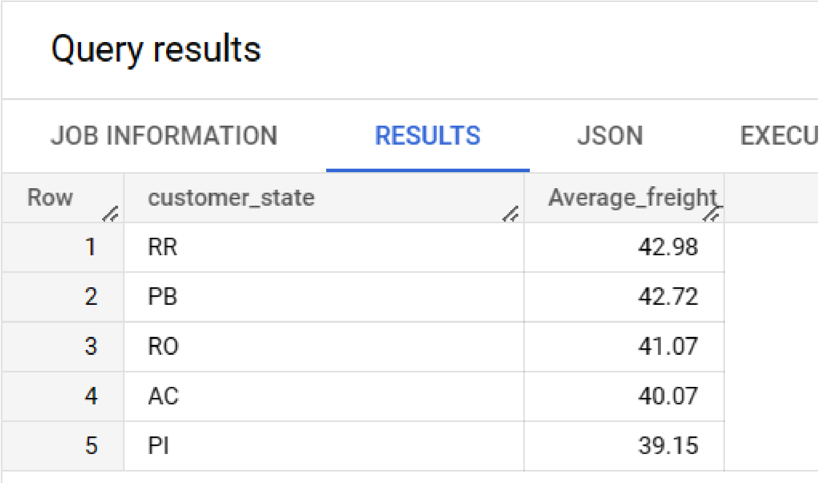


Row	State	Mean_Freight	time_to_delivery	diff_estimated_c
1	RR	42.98	2.0	2.0
2	PB	42.72	1.0	1.0
3	RO	41.07	0.0	0.0
4	AC	40.07	1.0	2.0
5	PI	39.15	1.0	1.0
6	MA	38.26	0.0	0.0
7	TO	37.25	0.0	0.0
8	SE	36.65	1.0	1.0
9	AL	35.84	0.0	0.0
10	PA	35.83	1.0	0.0

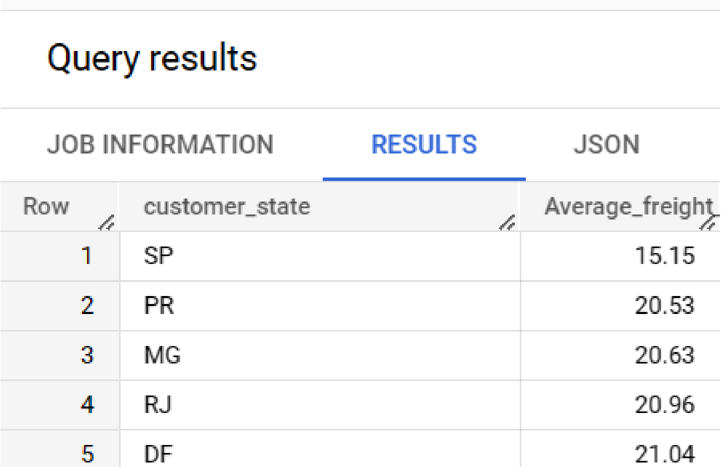
5.

```
            select
                c.customer_state,Round(avg(oi.freight_value),2) as Average_freight_value
        FROM
                `sqldemo-381616.Target_BusinessCase.Customers` c
        join
                `sqldemo-381616.Target_BusinessCase.Orders` o
        on
                 c.customer_id = o.customer_id
        join
                `sqldemo-381616.Target_BusinessCase.OrderItems` oi
        on
                o.order_id = oi.order_id
        group by
                c.customer_state
        order by
                Average_freight_value desc
        limit 5;

select
        c.customer_state,
        Round(avg(oi.freight_value),2) as Average_freight_value
FROM
```

```
        `sqldemo-381616.Target_BusinessCase.Customers` c
join
        `sqldemo-381616.Target_BusinessCase.Orders` o
on
        c.customer_id = o.customer_id
join
        `sqldemo-381616.Target_BusinessCase.OrderItems` oi
on
        o.order_id = oi.order_id
group by
        c.customer_state
order by
        Average_freight_value
limit 5;
```



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

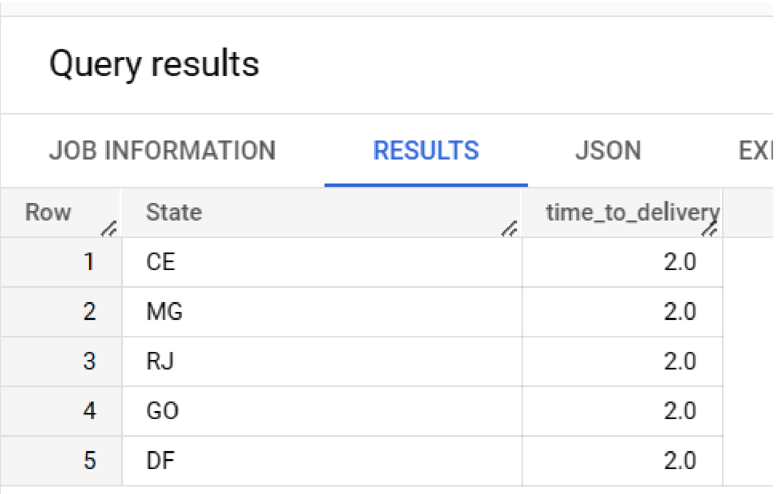


Row	customer_state	Average_freight
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

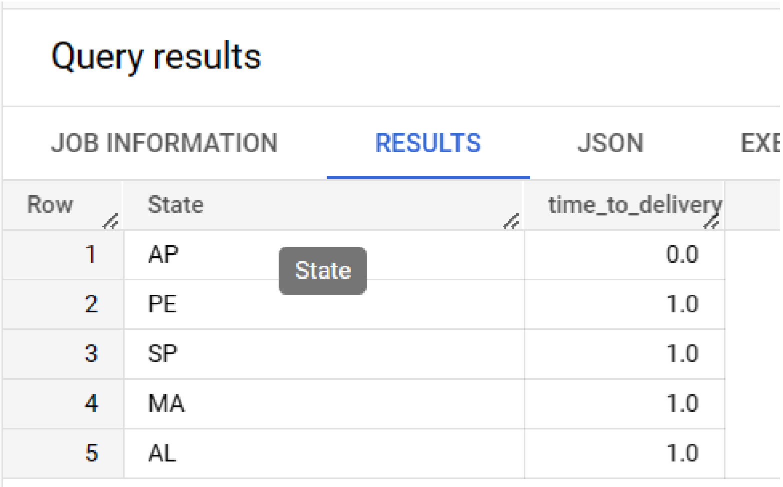
```
6. SELECT
        c.customer_state as State,
        Round(abs(avg(extract(Hour from o.order_purchase_timestamp)-
        extract(Hour from o.order_delivered_customer_date))),0) as time_to_delivery
FROM
```

```
        `sqldemo-381616.Target_BusinessCase.Customers` c
join
        `sqldemo-381616.Target_BusinessCase.Orders` o
on
        c.customer_id = o.customer_id
join
        `sqldemo-381616.Target_BusinessCase.OrderItems` oi
on
        o.order_id = oi.order_id
group by
        c.customer_state
order by
        time_to_delivery desc
limit 5;
```

```
SELECT
        c.customer_state as State,
        Round(abs(avg(extract(Hour from o.order_purchase_timestamp)-
        extract(hour from o.order_delivered_customer_date))),0) as time_to_delivery
FROM
        `sqldemo-381616.Target_BusinessCase.Customers` c
join
        `sqldemo-381616.Target_BusinessCase.Orders` o
on
        c.customer_id = o.customer_id
join
        `sqldemo-381616.Target_BusinessCase.OrderItems` oi
on
        o.order_id = oi.order_id
group by
        c.customer_state
order by
        time_to_delivery
limit 5;
```



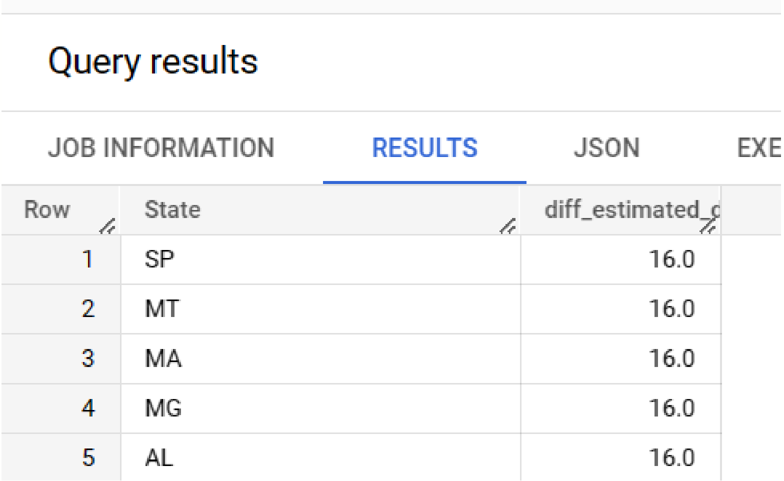
Row	State	time_to_delivery
1	CE	2.0
2	MG	2.0
3	RJ	2.0
4	GO	2.0
5	DF	2.0



Query results

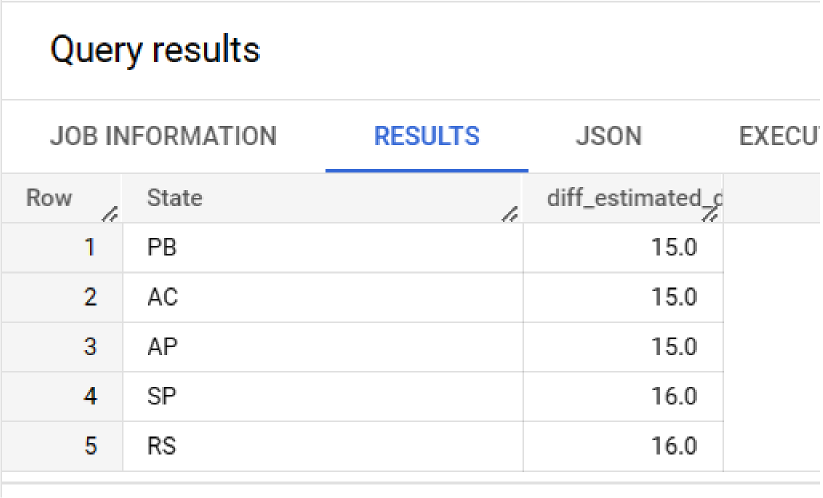
Row	State	time_to_delivery
1	AP	0.0
2	PE	1.0
3	SP	1.0
4	MA	1.0
5	AL	1.0

```
7. SELECT
          c.customer_state as State,
          Round(abs(avg(extract(hour from o.order_estimated_delivery_date)-
          extract(hour from o.order_delivered_customer_date))),0) as diff_estimated_delivery
  FROM
          `sqldemo-381616.Target_BusinessCase.Customers` c join `sqldemo-
          381616.Target_BusinessCase.Orders` o
on
          c.customer_id = o.customer_id
join
          `sqldemo-381616.Target_BusinessCase.OrderItems` oi
on
          o.order_id = oi.order_id
group by
          c.customer_state
order by
          diff_estimated_delivery desc
limit 5;
```



Query results

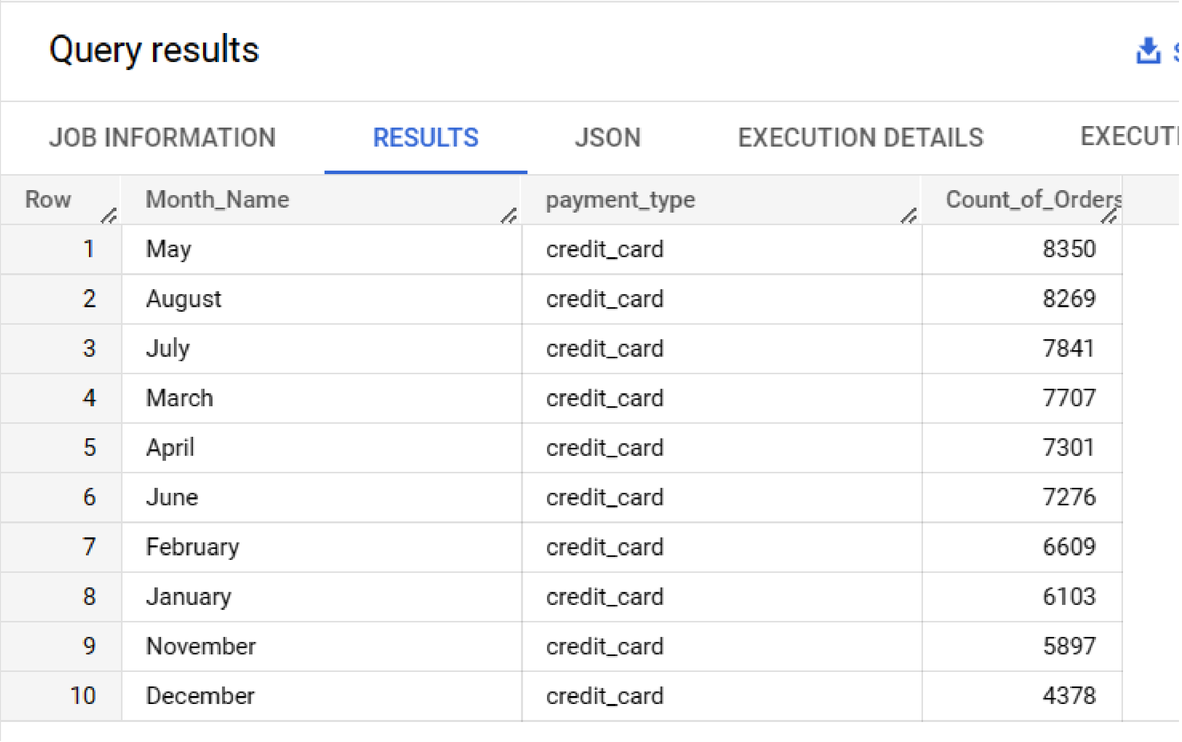
Row	State	diff_estimated_d
1	SP	16.0
2	MT	16.0
3	MA	16.0
4	MG	16.0
5	AL	16.0



Row	State	diff_estimated_c
1	PB	15.0
2	AC	15.0
3	AP	15.0
4	SP	16.0
5	RS	16.0

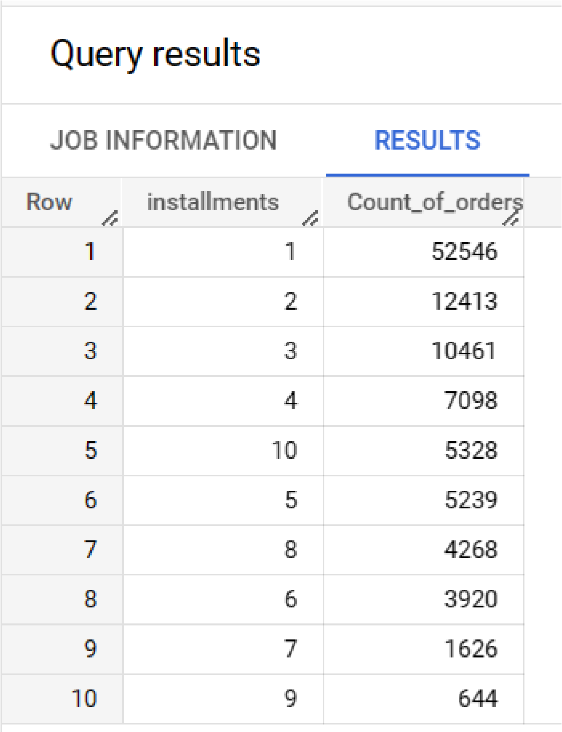
6.

```
1. SELECT
                FORMAT_DATETIME("%B",DATETIME (o.order_purchase_timestamp))
                as Month_Name,p.payment_type as payment_type, count(*) as Count_of_Orders
        FROM
                `sqldemo-381616.Target_BusinessCase.Orders` o
        left join
                `sqldemo-   381616.Target_BusinessCase.Payments` p
        on
                o.order_id = p.order_id
        group by
                payment_type,
                Month_Name
        order by
                Count_of_Orders desc
        LIMIT 10;
```



Row	Month_Name	payment_type	Count_of_Orders
1	May	credit_card	8350
2	August	credit_card	8269
3	July	credit_card	7841
4	March	credit_card	7707
5	April	credit_card	7301
6	June	credit_card	7276
7	February	credit_card	6609
8	January	credit_card	6103
9	November	credit_card	5897
10	December	credit_card	4378

2.
```
select
        p.payment_installments as installments,
        count(*) as Count_of_orders
FROM
        `sqldemo-381616.Target_BusinessCase.Orders` o
left join
        `sqldemo-381616.Target_BusinessCase.Payments` p
on
        o.order_id = p.order_id
group by
        installments
order by
        Count_of_orders desc
limit 10;
```



Query results

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

- Most of the credit card payments are having 3 or less installments, this information can be used to cross sell more products to people who use credit card.