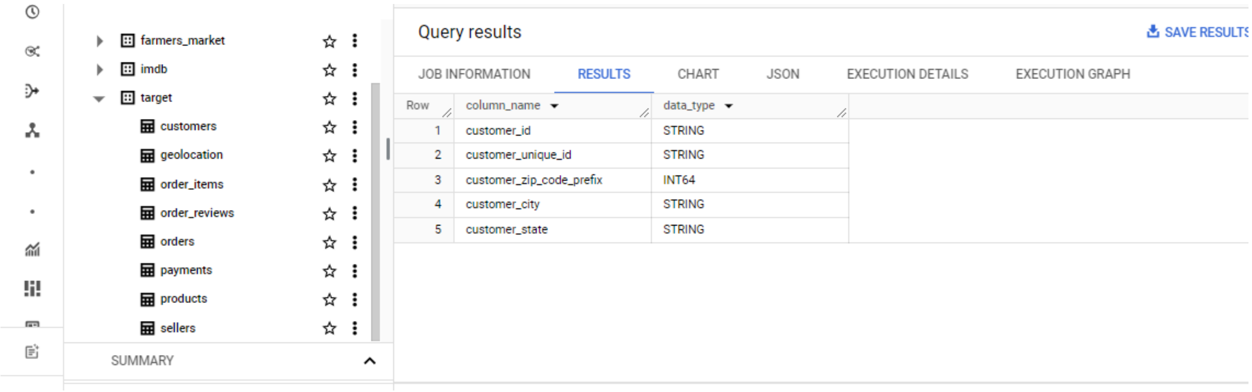
1.1. Data type of all columns in the "customers" table.

```
  select column_name,data_type

    from target.INFORMATION_SCHEMA.COLUMNS
    where table_name='customers';
```

Output-

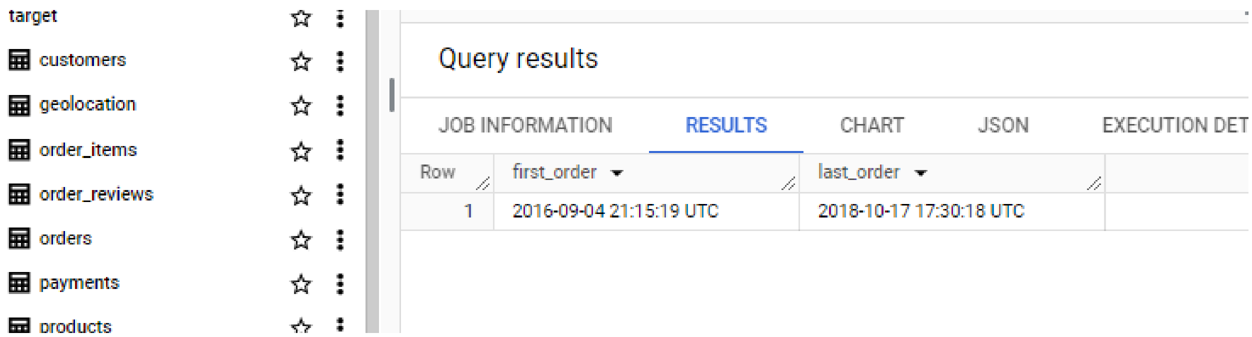


Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

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

```
select MIN(order_purchase_timestamp),
 MAX(order_purchase_timestamp)
    from `target.orders`;
```

Output-



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

Insights-

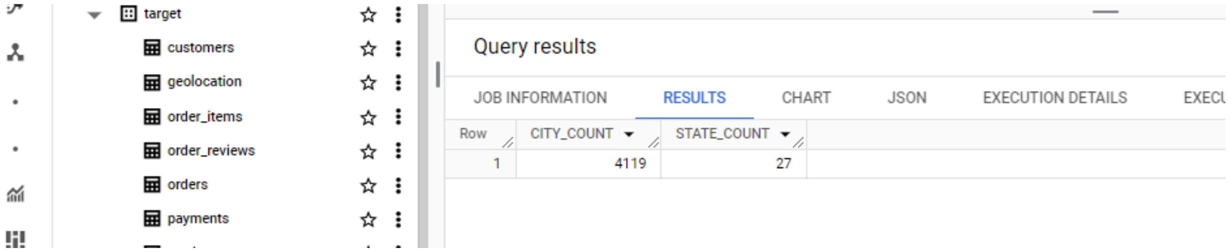
1. The very first_order was done on 2016-09-04 21:15:19.
2. The last order was done on 2018-10-17 17:30:18.

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

```
SELECT COUNT(DISTINCT customer_city)CITY_COUNT,COUNT(DISTINCT customer_state
)STATE_COUNT
    FROM `target.customers`
```

```
INNER JOIN `target.orders`
USING(CUSTOMER_ID);
```

Output



Query results

Row	CITY_COUNT	STATE_COUNT
1	4119	27

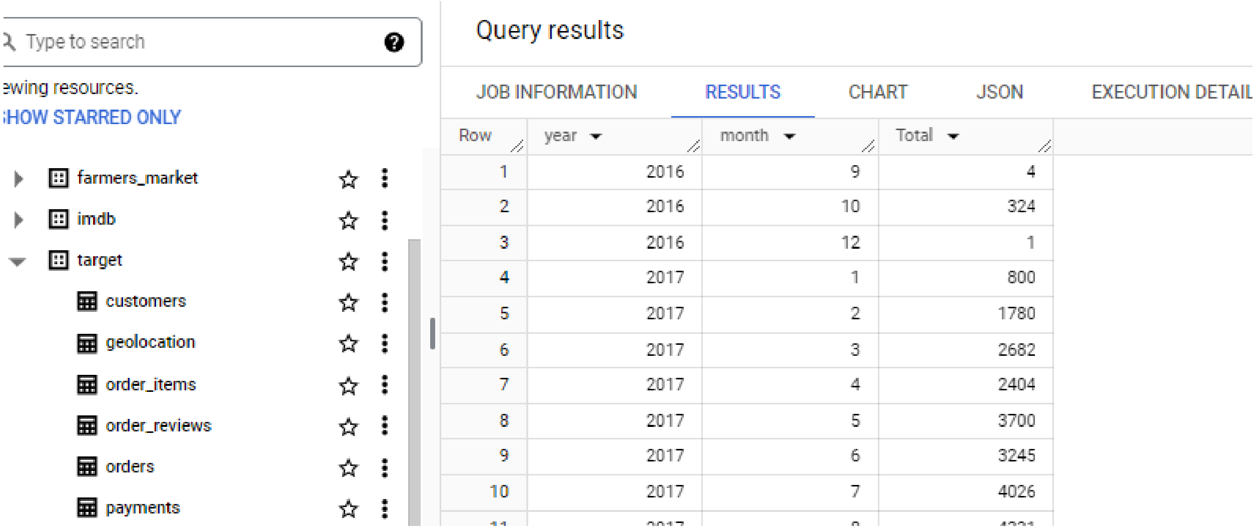
Insights-

1.No repetitive cities and states are included.

2.1 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)Total
  from `target.orders`
  group by year,month
  order by year,month;
```

Output-



Query results

Row	year	month	Total
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
11	2017	8	4331

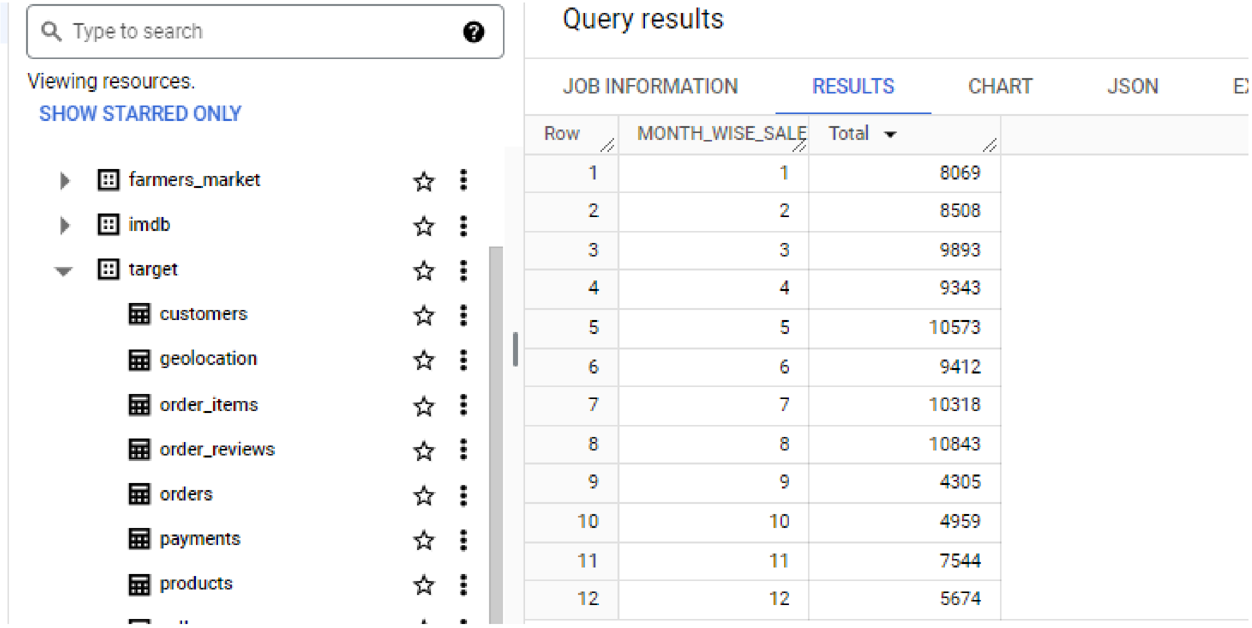
Insights-

1. From the given date we have only 3 months of data in the year 2016. In the next year We have all months data that makes it to total of 45101 orders and in the next year the orders are increased to 54011. So from the past 3 years the orders are increasing.

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

```
SELECT EXTRACT(MONTH FROM
order_purchase_timestamp)MONTH_WISE_SALES,COUNT(ORDER_ID)Total
  FROM `target.orders`
  GROUP BY MONTH_WISE_SALES
  ORDER BY MONTH_WISE_SALES;
```

Output-



Query results

Row	MONTH_WISE_SALE	Total
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

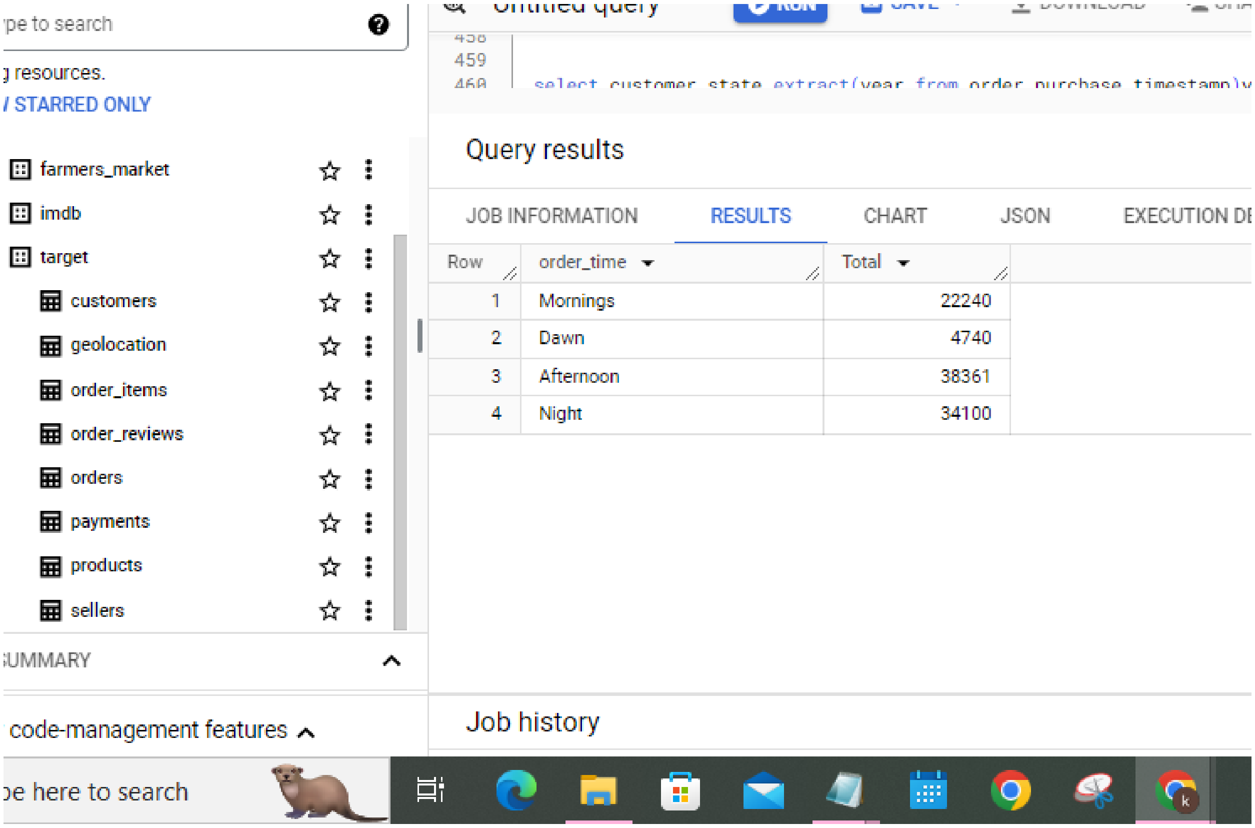
Insights-

1. In January and February we have decent number of orders . But in the next 6 months that is from march to august the orders are gone up. Similarly the orders are reduced from sept to dec.
2. So it is advisable to have good amount of storage in mar-aug season.

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

```
select order_time,count(order_id)Total
   from
   (SELECT order_id,
   CASE WHEN EXTRACT(TIME FROM order_purchase_timestamp) BETWEEN '00:00:00' and
'05:59:59' THEN 'Dawn'
     WHEN EXTRACT(TIME FROM order_purchase_timestamp) BETWEEN '06:00:00' and
'11:59:59' THEN 'Mornings'
     WHEN EXTRACT(TIME FROM order_purchase_timestamp) BETWEEN '12:00:00' and
'17:59:59' THEN 'Afternoon'
     else 'Night'end order_time
   FROM `target.orders`)tbl
   group by order_time;
```

Output-



```
select customer_state,extract(year from order_purchase_timestamp)y
```

Row	order_time	Total
1	Mornings	22240
2	Dawn	4740
3	Afternoon	38361
4	Night	34100

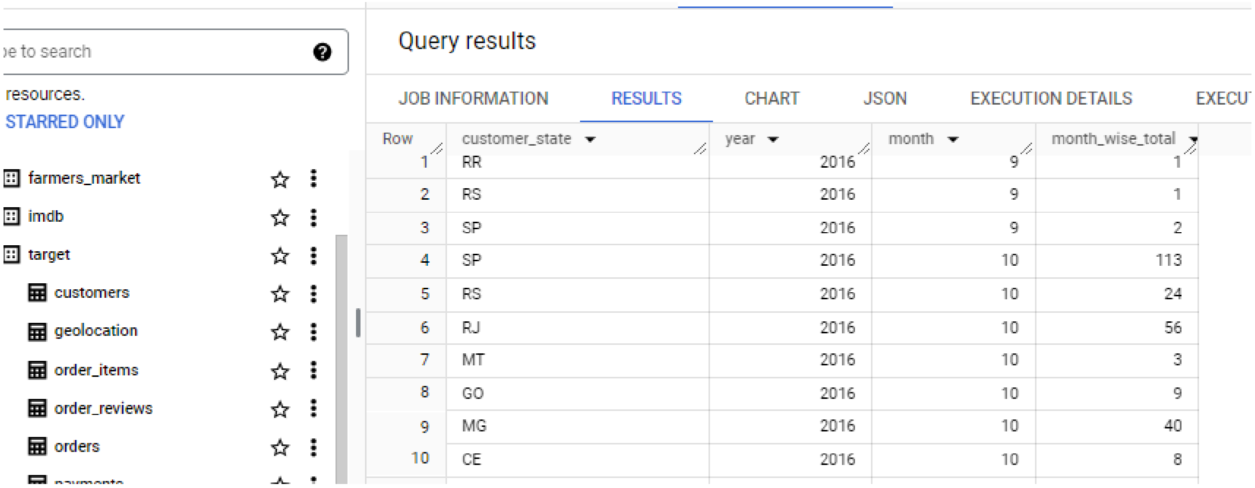
Insights-

1. The Brazilian customers placed more orders during afternoon .

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

```
select customer_state,extract(year from order_purchase_timestamp)year,
extract(month from order_purchase_timestamp)month,count(order_id)month_wise_total
from `target.orders`
inner join target.customers c
using (customer_id)
group by customer_state,year,month
order by year,month;
```

Output-



Row	customer_state	year	month	month_wise_total
1	RR	2016	9	1
2	RS	2016	9	1
3	SP	2016	9	2
4	SP	2016	10	113
5	RS	2016	10	24
6	RJ	2016	10	56
7	MT	2016	10	3
8	GO	2016	10	9
9	MG	2016	10	40
10	CE	2016	10	8

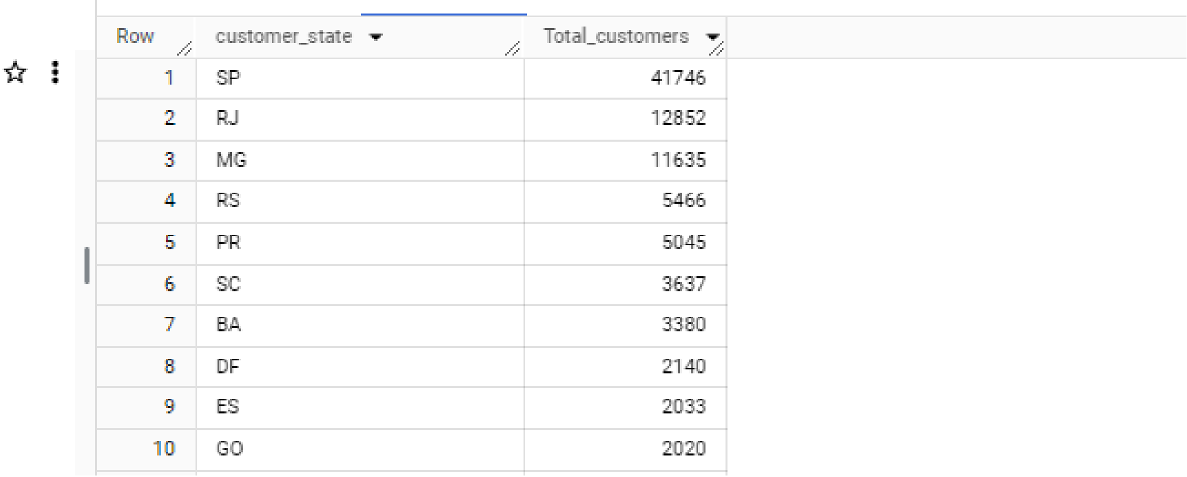
Insights-

1. A total of 654 orders have been received from the state SP alone in February 2017 which is the highest among all the states in a single month.
2. Increase the orders in the other state as well by giving discount, buy one get one free, combo Offers.

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

```
select customer_state,count(customer_id)Total_customers
from `target.customers`
group by customer_state
order by Total_customers desc;
```

Output-



Row	customer_state	Total_customers
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

Insights-

1.SP RJ and MG are the top three states in terms of number of orders.

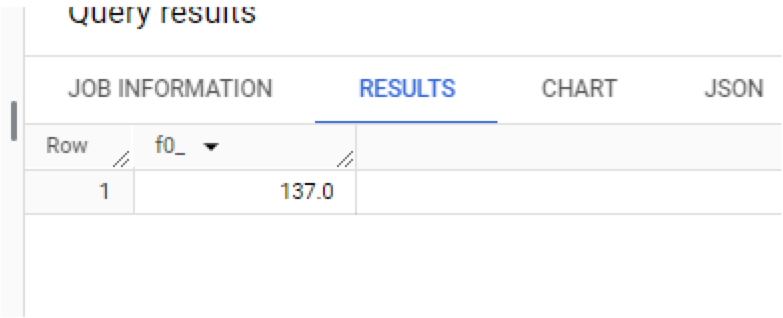
4.1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
with cte as
   (select extract(year from order_purchase_timestamp)year,sum(payment_value)cost
   from `target.payments`
   inner join`target.orders`
   using (order_id)
   where extract(month from order_purchase_timestamp) between 01 and 08
   group by year
   order by year),

   cte1 as
  (select year,cost,lead(cost) over(order by year)next_cost,(((lead(cost) over(order
by year)-cost)/cost)*100)change_in_percentage
  from cte)

  select round(change_in_percentage)
  from cte1
  limit 1;
```

Output-



Row	f0_
1	137.0

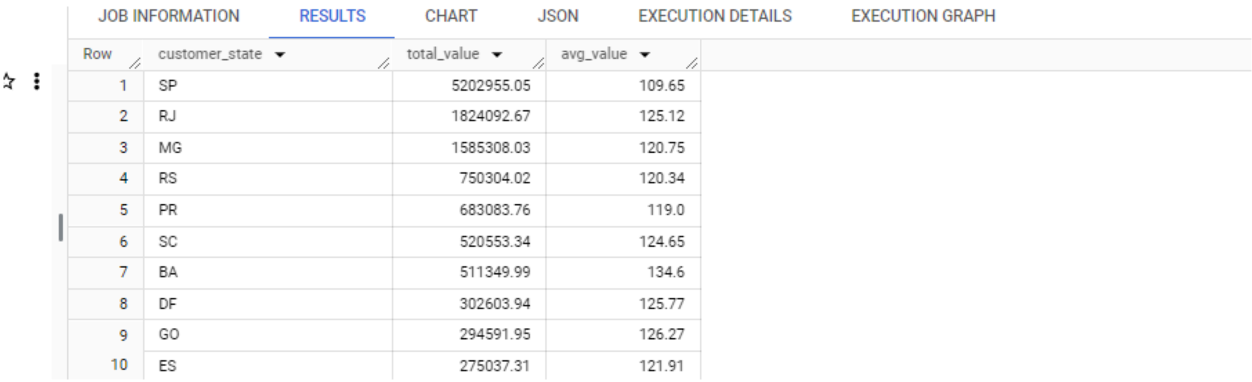
Insights-

1. There has been approximately 137% change in cost of orders in the year 2018 compared to 2017.
   For comparison only jan-aug months have been included from both the years.

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

```
  select customer_state,round(sum(price),2)total_value,round(avg(price),2)avg_value
  from `target.customers`
  inner join `target.orders` o
  using(customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by total_value desc,avg_value desc;
```

Output-



Row	customer_state	total_value	avg_value
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

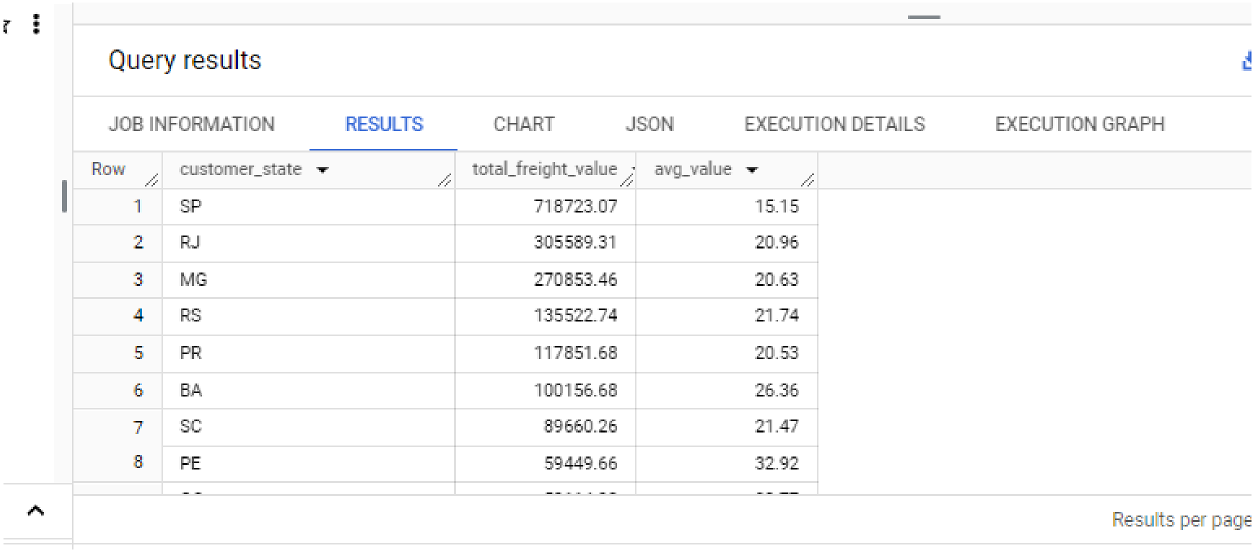
Insights-

1. SP, RJ and MG are top three states in terms of total value.
2. PB, AL and AC are top three states in terms of average value of an order.

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

```
  select
customer_state,round(sum(freight_value),2)total_freight_value,round(avg(freight_value
),2)avg_value
  from `target.customers`
  inner join `target.orders` o
  using (customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by total_freight_value desc,avg_value;
```

Output-



Row	customer_state	total_freight_value	avg_value
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92

Insights-

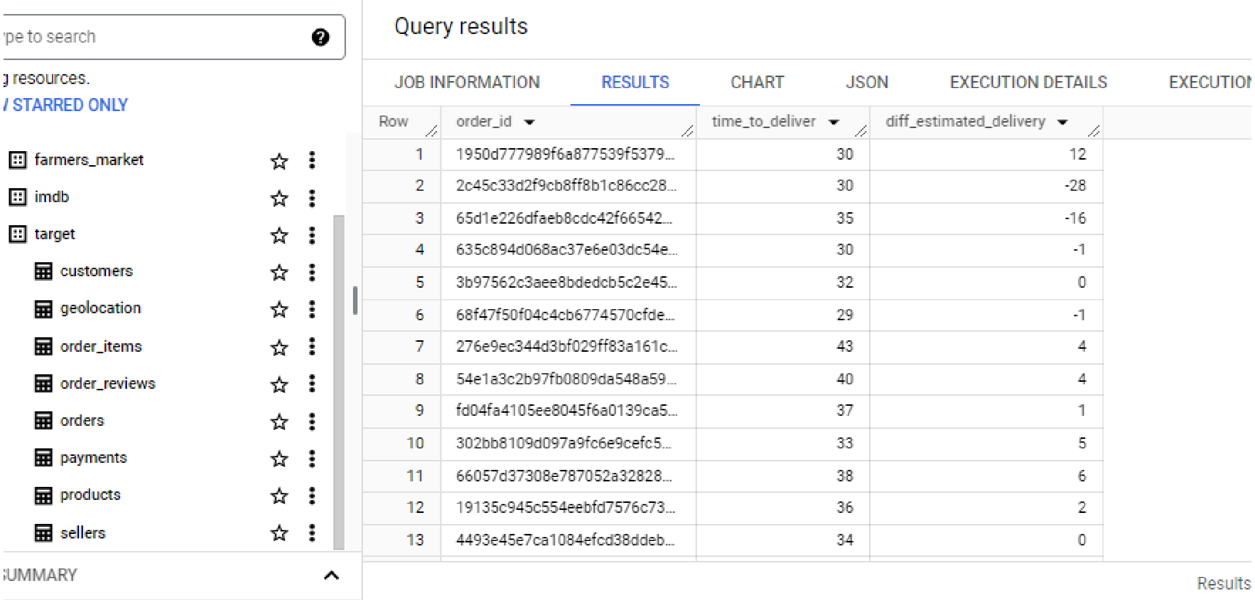
1. SP, RJ, MG are having more freight value because the orders from these states are also more.
2. RR, PB, RO are having more average freight value than all other states.

3. Prefer road and rail transport options which are cheaper than air transport to Reduce the freight charges.

5.1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

```
select order_id,date_diff(order_delivered_customer_date,
order_purchase_timestamp,day)time_to_deliver,
date_diff(order_delivered_customer_date,
order_estimated_delivery_date,day)diff_estimated_delivery
from `target.orders`;
```

Output-



Query results

Row	order_id	time_to_deliver	diff_estimated_delivery
1	1950d777989f6a877539f5379...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	-28
3	65d1e226dfaeb8cdc42f66542...	35	-16
4	635c894d068ac37e6e03dc54e...	30	-1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	-1
7	276e9ec344d3bf029ff83a161c...	43	4
8	54e1a3c2b97fb0809da548a59...	40	4
9	fd04fa4105ee8045f6a0139ca5...	37	1
10	302bb8109d097a9fc6e9cefc5...	33	5
11	66057d37308e787052a32828...	38	6
12	19135c945c554eebfd7576c73...	36	2
13	4493e45e7ca1084efcd38ddeb...	34	0

Insights-

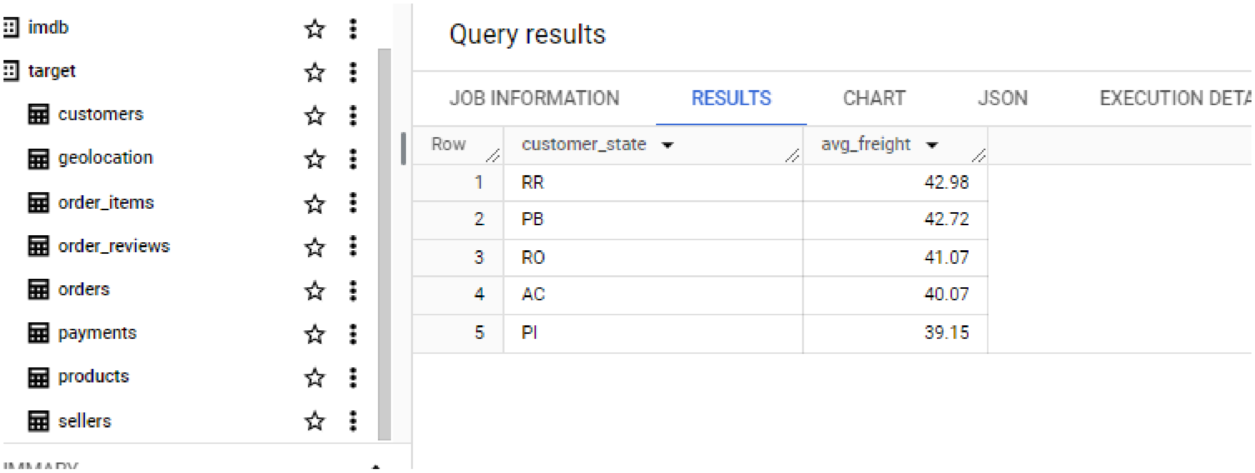
1. Some orders took more than expected date of delivery. So improve dispatch and Delivery management.

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

Top 5 Highest avg freight value states

```
  select customer_state,avg_freight
    from
  (select customer_state,round(avg(freight_value),2)avg_freight,dense_rank()
over(order by round(avg(freight_value),2) desc)rnk
  from `target.customers`
  inner join `target.orders` o
  using (customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by rnk)tbl
  where rnk<=5;
```

Output-



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

Insights-

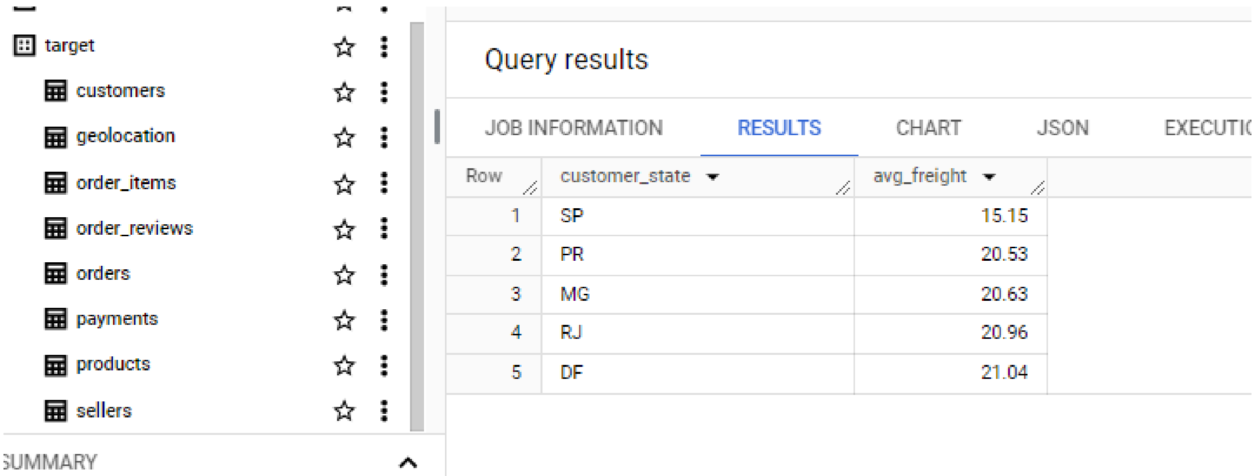
1.RR, PB, RO, AC, PI are the top 5 states that are having highest avg freight charges.

Top 5 lowest avg freight value states

```
 select customer_state,avg_freight

    from
  (select customer_state,round(avg(freight_value),2)avg_freight,dense_rank()
over(order by round(avg(freight_value),2))rnk
 from `target.customers`
 inner join `target.orders` o
 using (customer_id)
 inner join `target.order_items` oi
 on o.order_id=oi.order_id
 group by customer_state
 order by rnk)tbl
 where rnk<=5;
```

Output-



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

Insights-

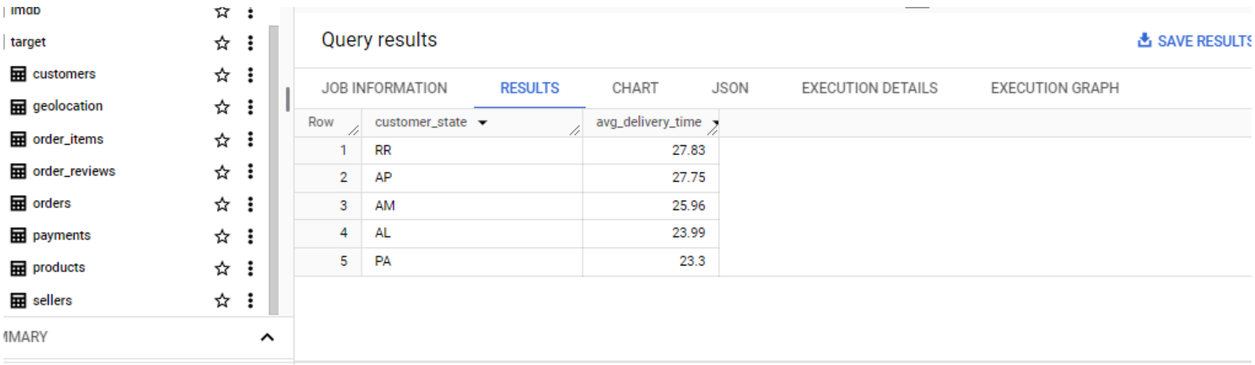
1. SP, PR, MG, RJ, DF are having lowest avg freight charges compared to other states.

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

Top_5_highest_avg_delivery_time

```
  select customer_state,avg_delivery_time
  from
  (select
customer_state,round(avg(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)),2)avg_delivery_time,
  dense_rank() over(order by
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)de
sc )rnk
  from `target.customers`
  inner join `target.orders` o
  using (customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by rnk)tbl
  where rnk<=5;
```

Output-



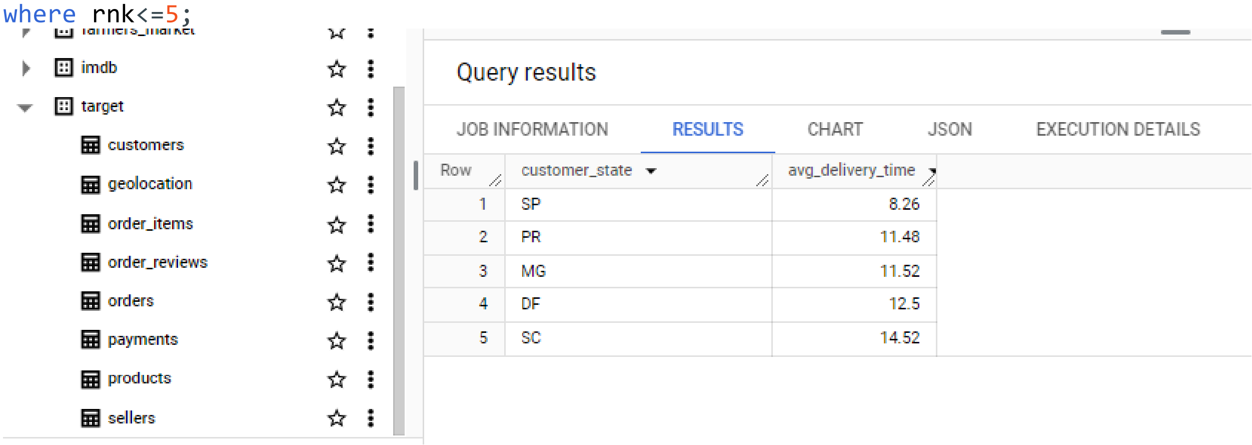
Row	customer_state	avg_delivery_time
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3

Insights-

1. RR, AP, AM, AL and PA are top 5 states in terms of taking more avg delivery time.

Top_5_lowest_avg_delivery_time

```
  select customer_state,avg_delivery_time
  from
  (select
customer_state,round(avg(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)),2)avg_delivery_time,
  dense_rank() over(order by
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)as
c )rnk
  from `target.customers`
  inner join `target.orders` o
  using (customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by rnk)tbl
where rnk<=5;
```



Row	customer_state	avg_delivery_time
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

Insights-

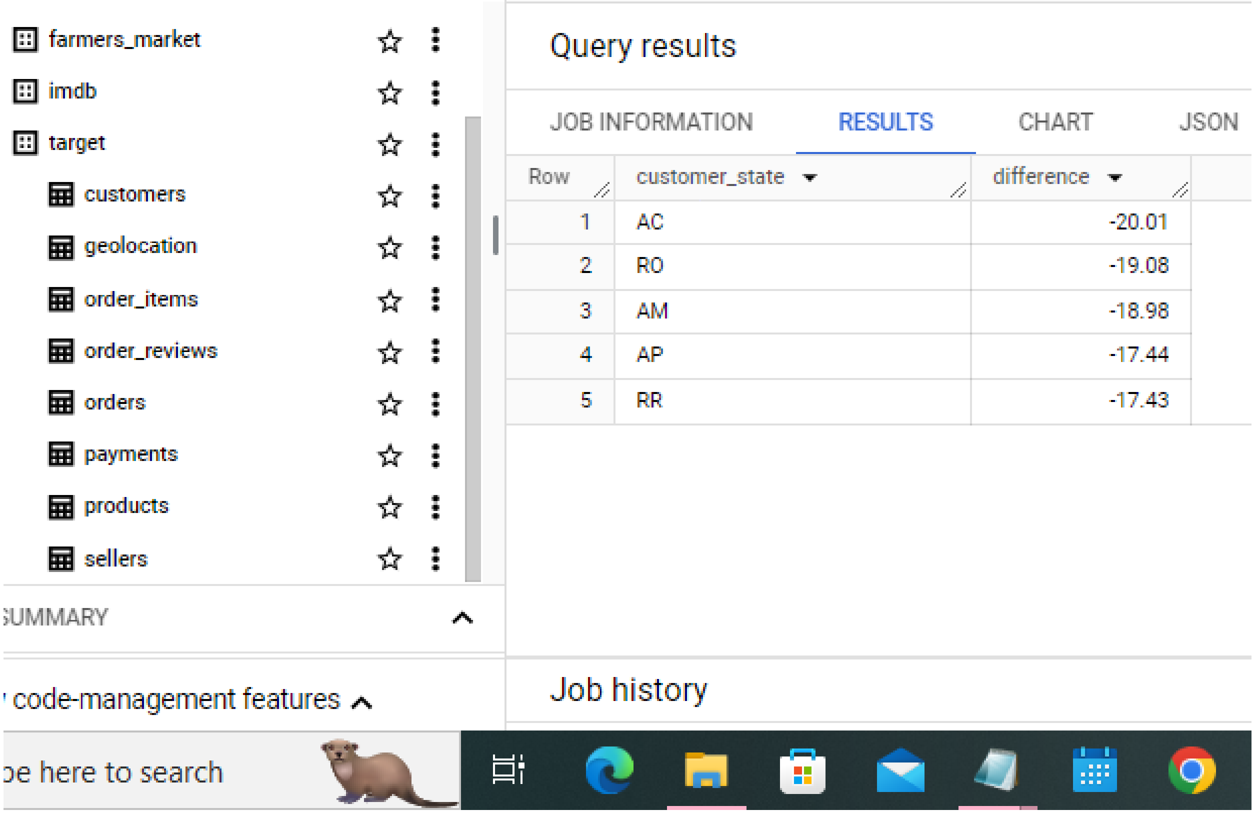
1. SP, PR, MG, DF and SC are top 5 states in terms of taking less avg delivery time.

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

top_5_fastest delivery states

```
  select customer_state,difference
  from
  (select
customer_state,round(avg(date_diff(order_delivered_customer_date,order_estimated_deli
very_date,day)),2)difference,
  dense_rank() over(order by
round(avg(date_diff(order_delivered_customer_date,order_estimated_delivery_date,day))
,2) asc)rnk
  from `target.customers`
  inner join  `target.orders` o
  using(customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by difference ) tbl
  where rnk<=5;
```

Output



Row	customer_state	difference
1	AC	-20.01
2	RO	-19.08
3	AM	-18.98
4	AP	-17.44
5	RR	-17.43

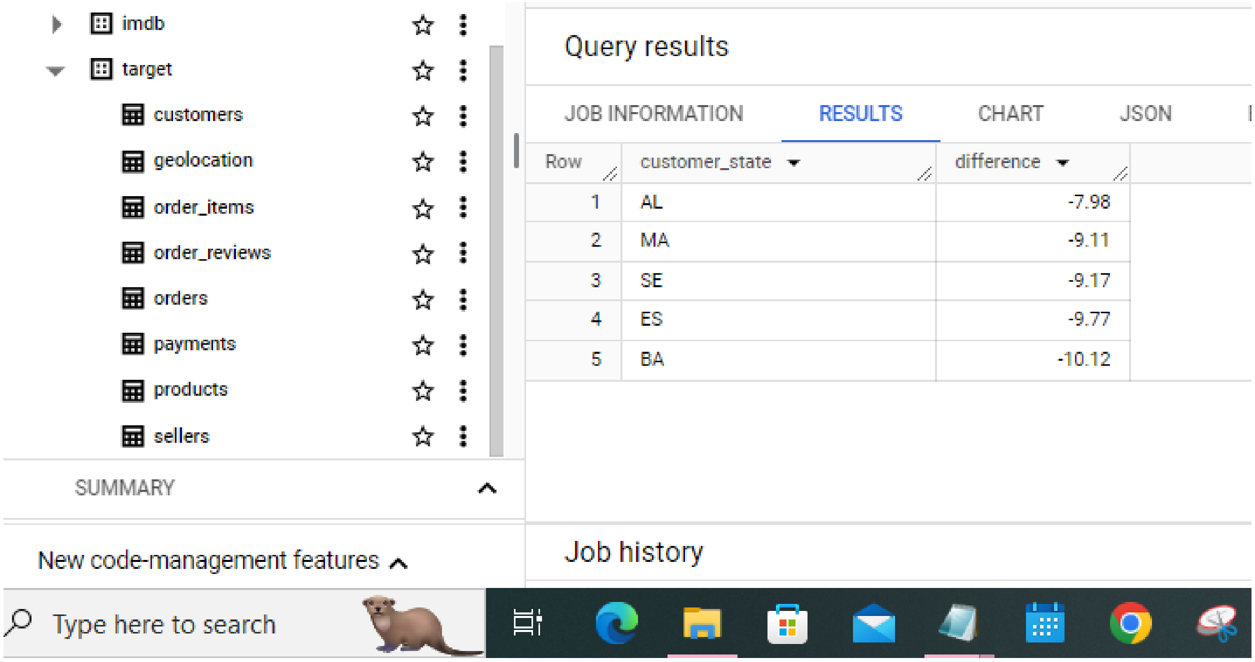
Insights-

1. Ac, RO, AM, AP, RR are the states where the order delivery is really fast as compared to the estimated date of delivery.

top_5_slowest delivery states

```
     select customer_state,difference
  from
  (select
customer_state,round(avg(date_diff(order_delivered_customer_date,order_estimated_deli
very_date,day)),2)difference,
  dense_rank() over(order by
round(avg(date_diff(order_delivered_customer_date,order_estimated_delivery_date,day))
,2) desc)rnk
  from `target.customers`
  inner join  `target.orders` o
  using(customer_id)
  inner join `target.order_items` oi
  on o.order_id=oi.order_id
  group by customer_state
  order by difference desc) tbl
  where rnk<=5;
```

Output-



Row	customer_state	difference
1	AL	-7.98
2	MA	-9.11
3	SE	-9.17
4	ES	-9.77
5	BA	-10.12

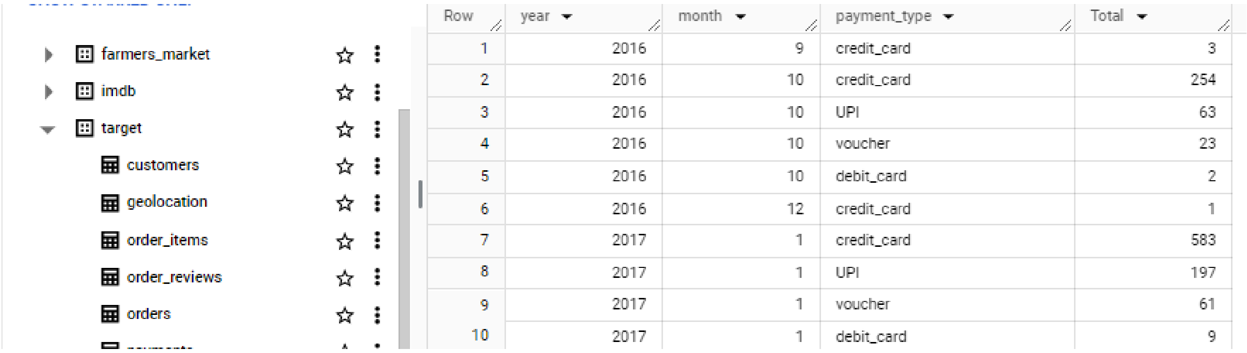
Insights-

1. AL, MA, SE, ES, BA are the states where the order delivery is slow among all the states as compared to the estimated date of delivery.

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

```
 select  extract(year from order_purchase_timestamp)year,extract(month from
order_purchase_timestamp)month,
  payment_type,count(order_id)Total
  from `target.orders`
  inner join `target.payments`
  using(order_id)
  group by year,month,payment_type
  order  by year,month;
```

Output-



Row	year	month	payment_type	Total
1	2016	9	credit_card	3
2	2016	10	credit_card	254
3	2016	10	UPI	63
4	2016	10	voucher	23
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	583
8	2017	1	UPI	197
9	2017	1	voucher	61
10	2017	1	debit_card	9

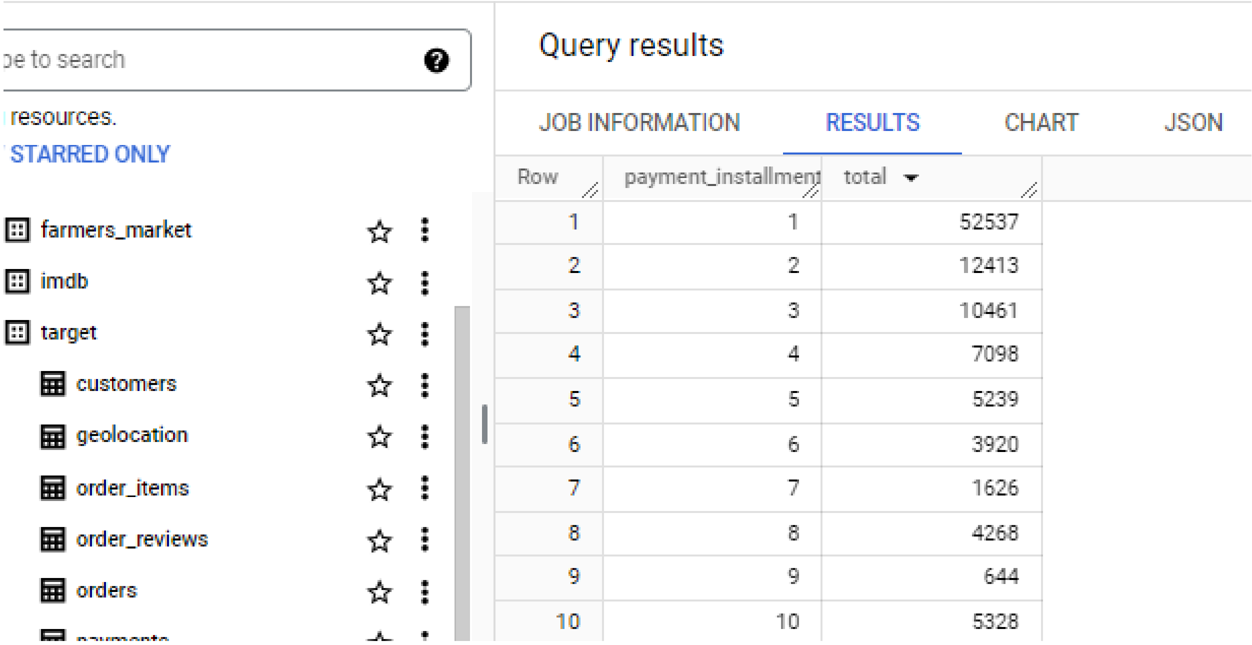
Insights-

1. From the data provided we can see that we have only 3 months(9,10,12) data in the year 2016.
2. Multiple payment options voucher, credit card, UPI, Debit card were used for making the payments.

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

```
select payment_installments,count( order_id)total
 from `target.payments`
 where payment_installments >0 and payment_value>0
 group by payment_installments;
```

Output-



Row	payment_installment	total
1	1	52537
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

Insights-

1. The output includes data about the no of installments and amount that is paid.