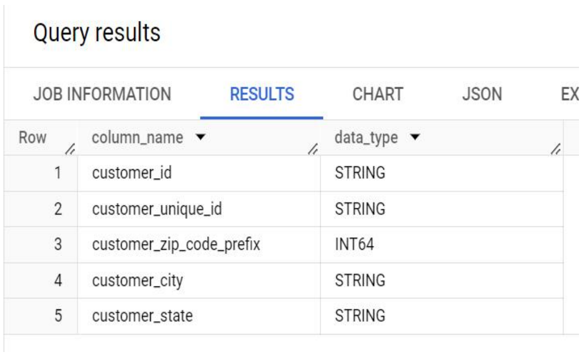
Business Case: Target SQL

The given DATASET is named as case_study

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

```
SELECT column_name, data_type
FROM case_study.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers';
```



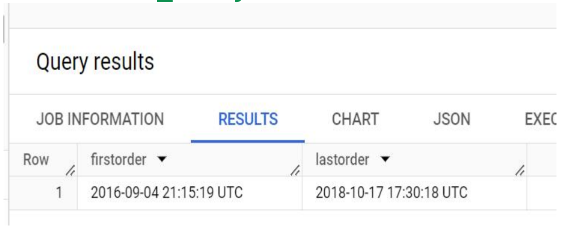
Query results

JOB INFORMATION RESULTS CHART JSON EX

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) as firstorder,
 max(order_purchase_timestamp) as lastorder
from `case_study.orders`
```



Query results

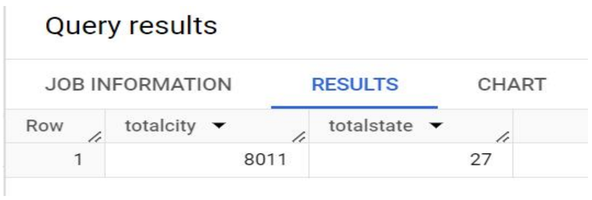
JOB INFORMATION RESULTS CHART JSON EXEC

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

The time range is between 2016-09-04 to 2018-10-17

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

```
select count(distinct geolocation_city) as totalcity, count(distinct
geolocation_state) as totalstate
from `case_study.geolocation`
```

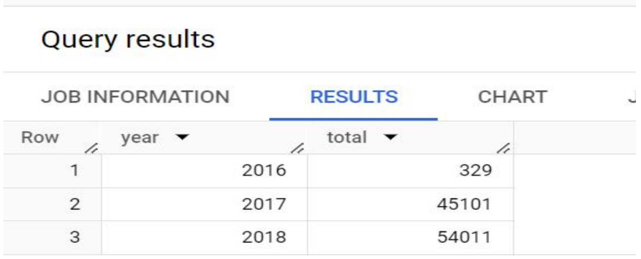

Query results

Row	totalcity	totalstate
1	8011	27

The above query gives the number of cities as totalcity and total number of states as totalstate, that ordered from 2016-09-04 to 2018-10-17.

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

```
select extract(year from order_purchase_timestamp) as year,
count(order_id) as total
from `case_study.orders`
group by 1
order by 1
```

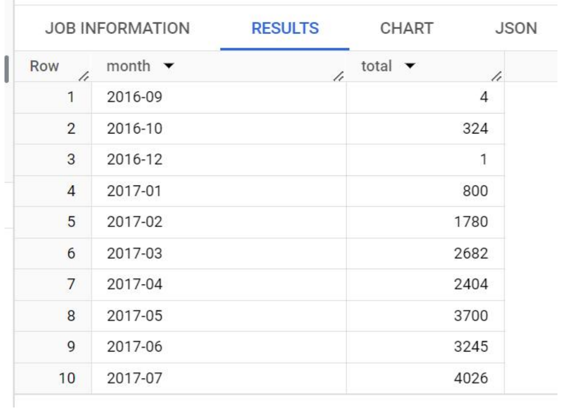

Query results

Row	year	total
1	2016	329
2	2017	45101
3	2018	54011

Total no. of orders each year is given as total. As the code says, the no. Of orders over the years has increased from 329 orders in 2016 to 54011 orders in 2018.

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

```
select FORMAT_DATETIME('%Y-%m', order_purchase_timestamp) as month,
count(order_id) as total
from `case_study.orders`
group by 1
order by 1
```

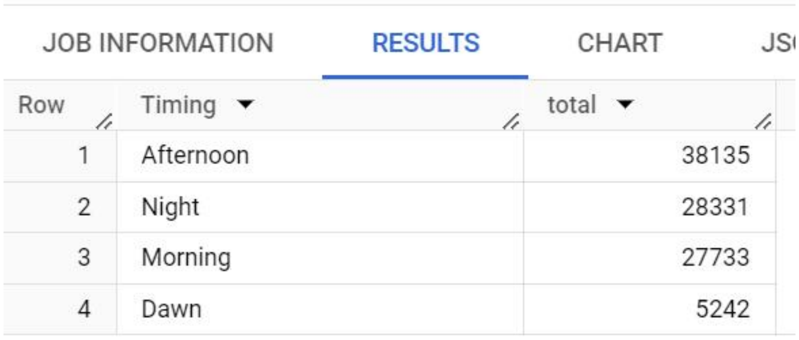


Row	month	total
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026

The initial months of the year has a decent amount of orders. The no. Of orders stays at it's peaks from the month of March till August. The no. Of orders gradually decreases in the later months of the year, i.e, from around September.

2.3 During what time of the day, do the Brazilian customers mostly place their orders?

```
select case when Hour between '00' and '06' then 'Dawn'
               when Hour between '07' and '12' then 'Morning'
               when Hour between '13' and '18' then 'Afternoon'
               when Hour between '19' and '23' then 'Night'
end as Timing, sum(totalorders) total
from (SELECT format_datetime('%H' , order_purchase_timestamp) as Hour,
count(order_id) as totalorders
FROM `case_study.orders`
group by 1
order by 2 DESC)
group by 1
order by 2 desc
```

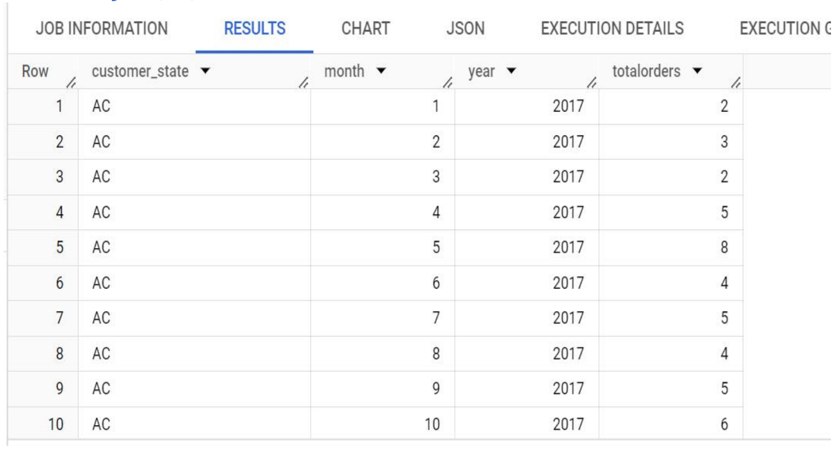


Row	Timing	total
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

The number of orders placed peaks during the Afternoon.

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

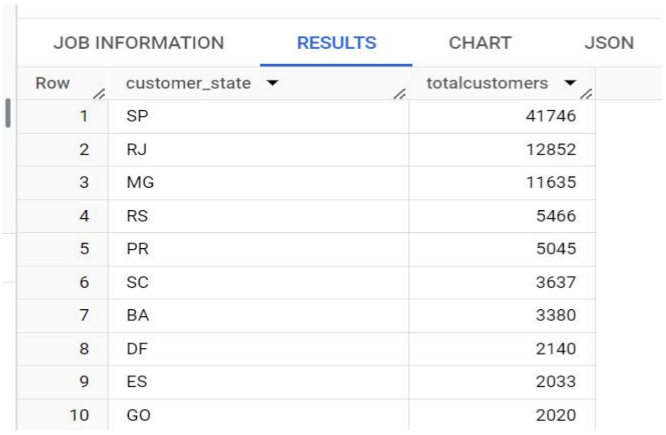
```
SELECT c.customer_state, extract(month from order_purchase_timestamp)
as month,extract(year from order_purchase_timestamp) as year,
count(order_id) as totalorders
from `case_study.customers` c
join `case_study.orders` o
on c.customer_id = o.customer_id
group by 1,2,3
order by 1,3,2
```



Row	customer_state	month	year	totalorders
1	AC	1	2017	2
2	AC	2	2017	3
3	AC	3	2017	2
4	AC	4	2017	5
5	AC	5	2017	8
6	AC	6	2017	4
7	AC	7	2017	5
8	AC	8	2017	4
9	AC	9	2017	5
10	AC	10	2017	6

3.2 How are the customers distributed across all the states?

```
select customer_state, count(distinct customer_id) as totalcustomers
from `case_study.customers`
group by 1
order by 2 desc
```

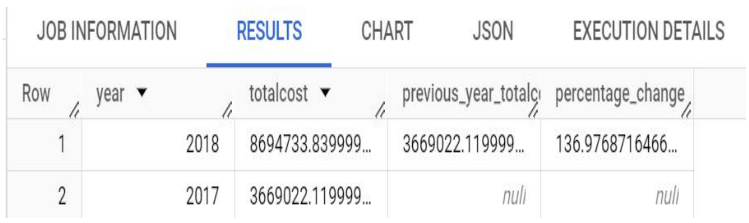


JOB INFORMATION | RESULTS | CHART | JSON

Row	customer_state	totalcustomers
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

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

```
with final as (select year, sum(total)as totalcost
from (SELECT extract(year from o.order_purchase_timestamp) as
year,extract(month from o.order_purchase_timestamp) as month,
 sum(payment_value) as total
from `case_study.orders` o
join `case_study.payments` p
on o.order_id=p.order_id
where extract(month from o.order_purchase_timestamp) between 01 and 08
group by 1,2)
group by year)
select year, totalcost,
  lag(totalcost) over (order by year) as previous_year_totalcost,
  ((totalcost - lag(totalcost) over (order by year)) / lag(totalcost)
over (order by year)) * 100 as percentage_change
from final
```



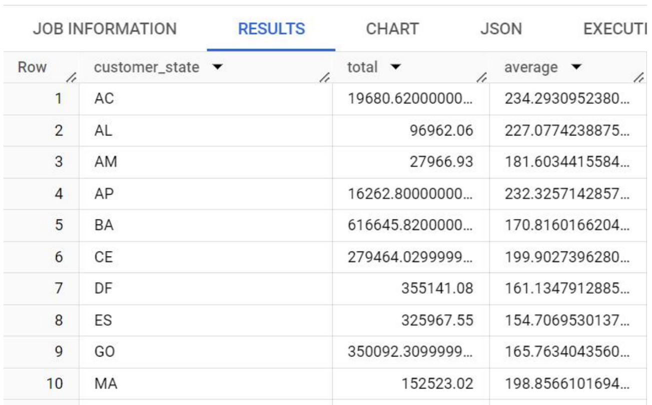
JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	year	totalcost	previous_year_totalc	percentage_change
1	2018	8694733.839999...	3669022.119999...	136.9768716466...
2	2017	3669022.119999...	null	null

The above code shows that the % increase in the cost of orders from year 2017 to 2018 including months between Jan to Aug only is 136.98%.

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

```
select c.customer_state, sum(oi.payment_value) as total,
avg(oi.payment_value) as average
from `case_study.payments` oi
join `case_study.orders` o
on oi.order_id = o.order_id
join `case_study.customers` c
on o.customer_id = c.customer_id
group by 1
order by 1
```

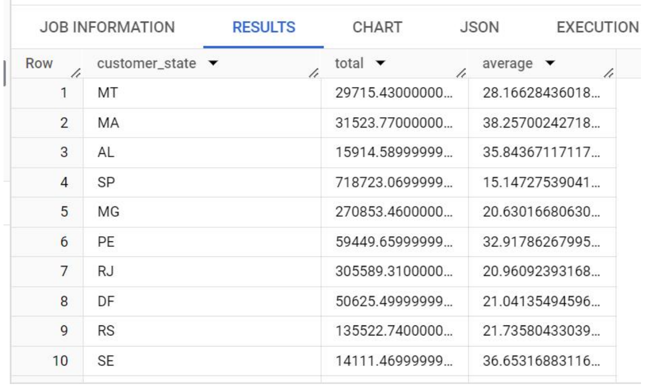


Row	customer_state	total	average
1	AC	19680.62000000...	234.2930952380...
2	AL	96962.06	227.0774238875...
3	AM	27966.93	181.6034415584...
4	AP	16262.80000000...	232.3257142857...
5	BA	616645.8200000...	170.8160166204...
6	CE	279464.0299999...	199.9027396280...
7	DF	355141.08	161.1347912885...
8	ES	325967.55	154.7069530137...
9	GO	350092.3099999...	165.7634043560...
10	MA	152523.02	198.8566101694...

The above code displays the total and average price based on each state.

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

```
select c.customer_state, sum(oi.freight_value) as total,
avg(oi.freight_value) as average
from `case_study.customers` c
join `case_study.orders` o
on c.customer_id = o.customer_id
join `case_study.order_items` oi
on o.order_id = oi.order_id
group by 1
```

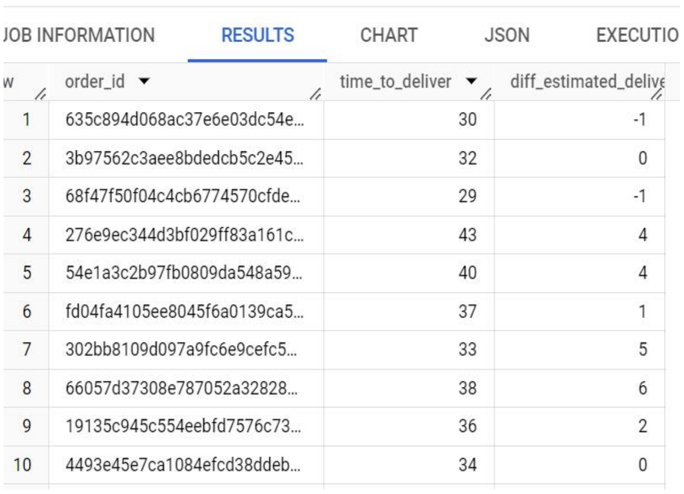


Row	customer_state	total	average
1	MT	29715.43000000...	28.16628436018...
2	MA	31523.77000000...	38.25700242718...
3	AL	15914.58999999...	35.84367117117...
4	SP	718723.0699999...	15.14727539041...
5	MG	270853.4600000...	20.63016680630...
6	PE	59449.65999999...	32.91786267995...
7	RJ	305589.3100000...	20.96092393168...
8	DF	50625.49999999...	21.04135494596...
9	RS	135522.7400000...	21.73580433039...
10	SE	14111.46999999...	36.65316883116...

The total and average of order freight for each state is given above in the column names total and average

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  order_id, date_diff(order_delivered_customer_date,
order_purchase_timestamp, day) as time_to_deliver,
date_diff(order_delivered_customer_date, order_estimated_delivery_date,
day) as diff_estimated_delivery
from `case_study.orders`
where order_status = 'delivered'
```

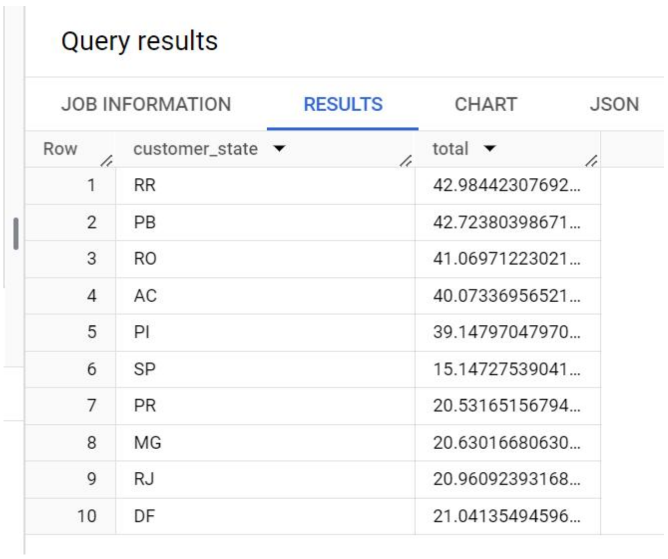


w	order_id	time_to_deliver	diff_estimated_delive
1	635c894d068ac37e6e03dc54e...	30	-1
2	3b97562c3aee8bdedcb5c2e45...	32	0
3	68f47f50f04c4cb6774570cfde...	29	-1
4	276e9ec344d3bf029ff83a161c...	43	4
5	54e1a3c2b97fb0809da548a59...	40	4
6	fd04fa4105ee8045f6a0139ca5...	37	1
7	302bb8109d097a9fc6e9cefc5...	33	5
8	66057d37308e787052a32828...	38	6
9	19135c945c554eebfd7576c73...	36	2
10	4493e45e7ca1084efcd38ddeb...	34	0

The time_to_deliver column has the number of days it took for orders to get delivered. The negative values in the `diff_estimated_delivery` column indicate delivery before estimated time.

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

```
(select c.customer_state, avg(oi.freight_value) as total
from `case_study.order_items` oi
join `case_study.orders` o
on oi.order_id = o.order_id
join `case_study.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 DESC
limit 5)
union all
(select c.customer_state, avg(oi.freight_value) as total
from `case_study.order_items` oi
join `case_study.orders` o
on oi.order_id = o.order_id
join `case_study.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 ASC
limit 5)
```


Query results

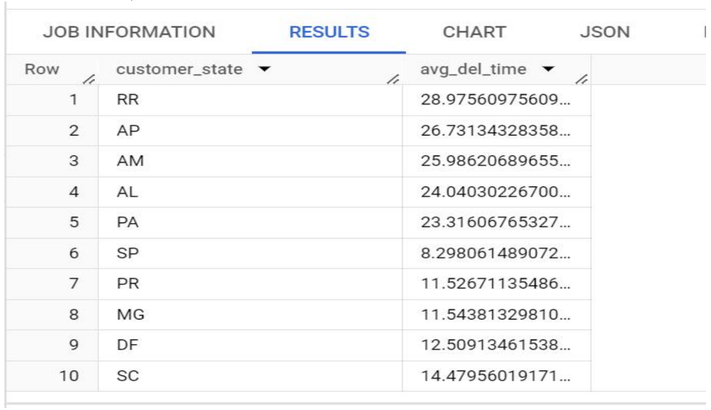
JOB INFORMATION | RESULTS | CHART | JSON

Row	customer_state	total
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...
6	SP	15.14727539041...
7	PR	20.53165156794...
8	MG	20.63016680630...
9	RJ	20.96092393168...
10	DF	21.04135494596...

The top 5 rows from the above output displays the top 5 states with the highest average freight values and the bottom 5 rows displays 5 states with lowest average freight values

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

```
(select c.customer_state, avg(date_diff(order_delivered_customer_date,
order_purchase_timestamp, day)) as avg_del_time
 from `case_study.customers` c
 join `case_study.orders` o
 on c.customer_id = o.customer_id
 group by 1
 order by 2 DESC
 limit 5)
UNION ALL
(select c.customer_state, avg(date_diff(order_delivered_customer_date,
order_purchase_timestamp, day)) as avg_del_time
 from `case_study.customers` c
 join `case_study.orders` o
 on c.customer_id = o.customer_id
 group by 1
 order by 2 asc
 limit 5)
```

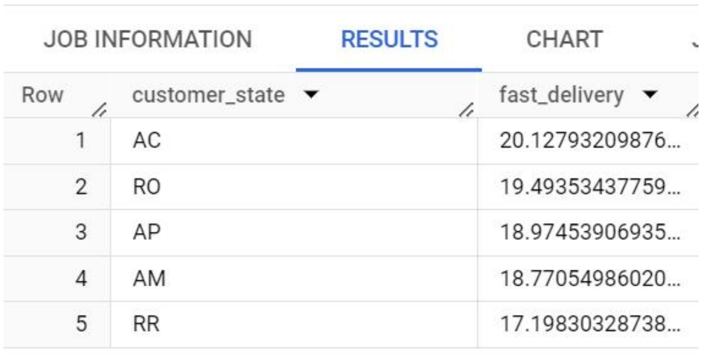

JOB INFORMATION | RESULTS | CHART | JSON

Row	customer_state	avg_del_time
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...
6	SP	8.298061489072...
7	PR	11.52671135486...
8	MG	11.54381329810...
9	DF	12.50913461538...
10	SC	14.47956019171...

The top 5 rows from the above output displays the top 5 states with the highest average delivery time and the bottom 5 rows displays 5 states with lowest average 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.

```
with final as (select c.customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,
day)) as avg_del_time,
       avg(date_diff(order_estimated_delivery_date,
order_purchase_timestamp, day)) as avg_est_time
from `case_study.customers` c
join `case_study.orders` o
on c.customer_id = o.customer_id
group by 1)
select customer_state, final.avg_est_time - final.avg_del_time as
fast_delivery
from final
order by 2 desc
Limit 5
```

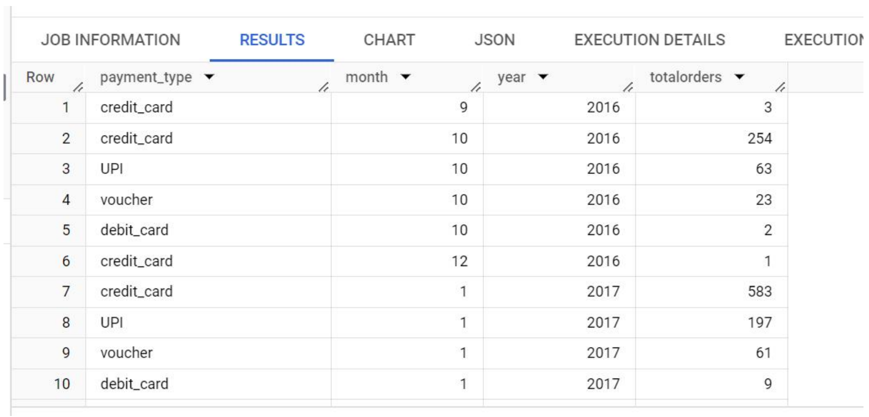

JOB INFORMATION | RESULTS | CHART

Row	customer_state	fast_delivery
1	AC	20.12793209876...
2	RO	19.49353437759...
3	AP	18.97453906935...
4	AM	18.77054986020...
5	RR	17.19830328738...

The column fast_delivery displays the number days the order delivered before estimated dates, I.e how many days faster the order got delivered. The table shows the top 5 states with fastest deliveries, from 20 days to 17 days faster deliveries of orders.

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

```
SELECT  p.payment_type, extract(month from order_purchase_timestamp) as
month,
extract(year from order_purchase_timestamp) as year, count(o.order_id)
as totalorders
from `case_study.orders` o
join `case_study.payments` p
on o.order_id = p.order_id
group by 3,2,1
order by 3,2
```



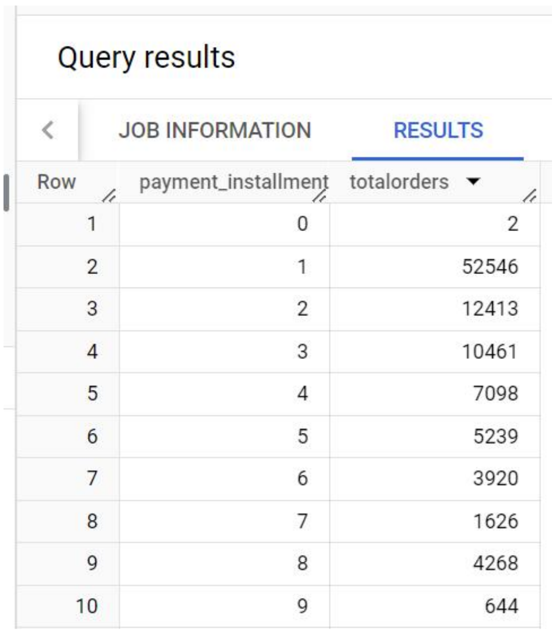
JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTIO

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

The above table displays the month on month orders by payment types that were used to place the orders.

6.2 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) as totalorders
from `case_study.orders` o
join `case_study.payments` p
on o.order_id = p.order_id
group by 1
order by 1
```



Query results

JOB INFORMATION | RESULTS

Row	payment_installment	totalorders
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

The table displays the count of orders placed for each distinct value of payment installments .

INSIGHTS AND RECOMMENDATION

- The time range of orders falls between September 4, 2016, and October 17, 2018. This indicates the duration of data collection for orders.

- The number of cities and states from which customers ordered during the specified period can provide insights into the geographic reach of the business. The states, SP, RJ, MG has the most customers. Focus on expanding marketing efforts or logistics networks in these states.

- There's a clear increasing trend in the number of orders placed over the years, indicating potential business growth. Invest in scaling up operations, optimizing inventory management, and improving customer service to accommodate the increasing demand.

- Allocate resources more efficiently during the initial months of the year, i.e, January to August and strategize promotional activities during slower months to stimulate sales.

- Schedule promotional emails or ads to coincide with peak order times i.e, Afternoon and Night and ensure customer service availability during high-traffic periods.

- Invest in faster shipping options to reduce delivery times and meet customer expectations in the states with slow delivery time.