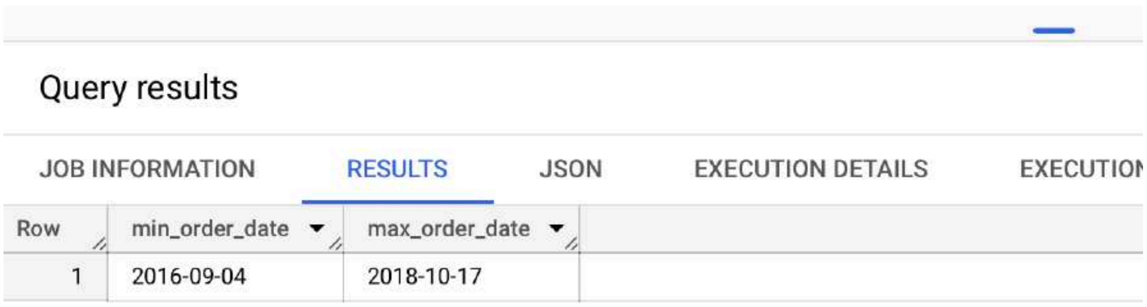
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

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

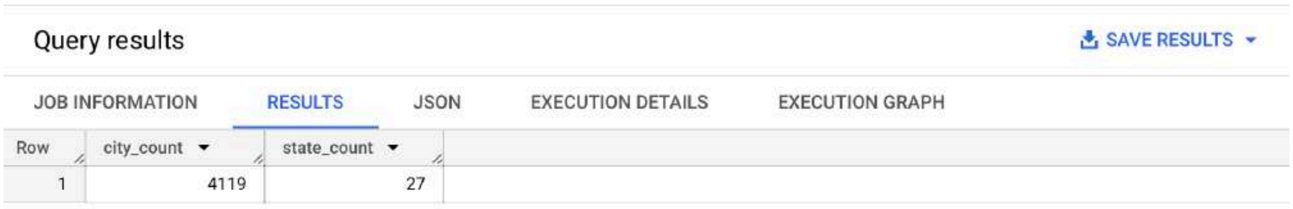
```
select
distinct (select min(date(order_purchase_timestamp)) from `TargetCaseStudy.orders`) as
min_order_date,
(select max(date(order_purchase_timestamp)) from `TargetCaseStudy.orders`) as
max_order_date,
from `TargetCaseStudy.orders`
```



Row	min_order_date	max_order_date
1	2016-09-04	2018-10-17

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

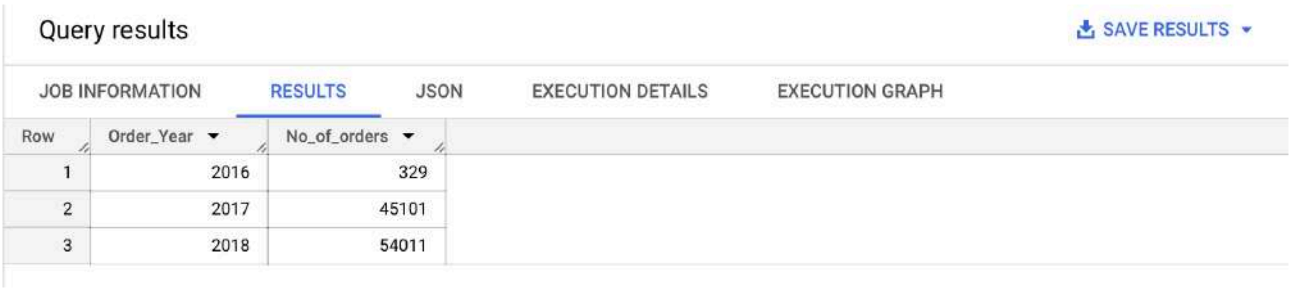
```
select count(distinct c.customer_city) as city_count, count(distinct c.customer_state) as
state_count from `TargetCaseStudy.customers` c
join `TargetCaseStudy.orders` o
on c.customer_id=o.customer_id
where (order_purchase_timestamp) >= '2016-09-04' and date(order_purchase_timestamp) <=
‘2018-10-17';
```



Row	city_count	state_count
1	4119	27

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 Order_Year,
select extract(MONTH from order_purchase_timestamp) as Order_Month,
count(*) as No_of_orders from `TargetCaseStudy.orders`
GROUP BY Order_Year, Order_Month,
order by Order_Year, Order_Month,
```


Query results

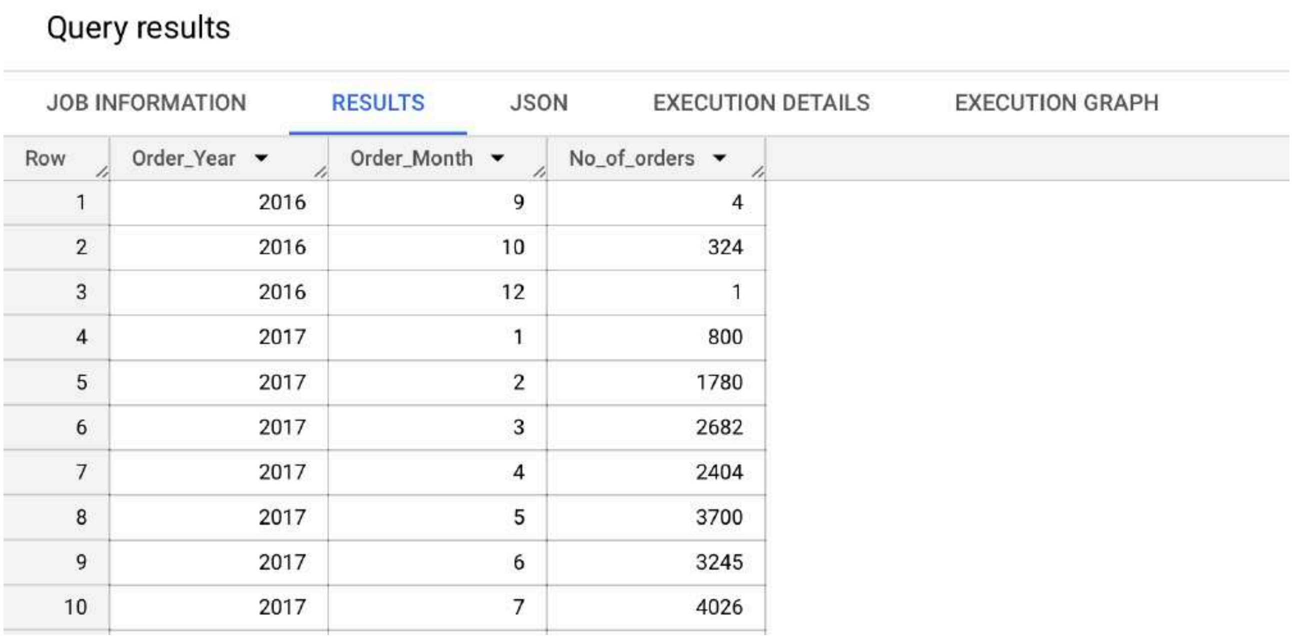
SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Order_Year	No_of_orders
1	2016	329
2	2017	45101
3	2018	54011

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

```
SELECT
extract(YEAR from order_purchase_timestamp) as Order_Year,
extract(MONTH from order_purchase_timestamp) as Order_Month,
count(*) as No_of_orders
from `TargetCaseStudy.orders`
GROUP BY Order_Year, Order_Month
order by Order_Year, Order_Month;
```


Query results

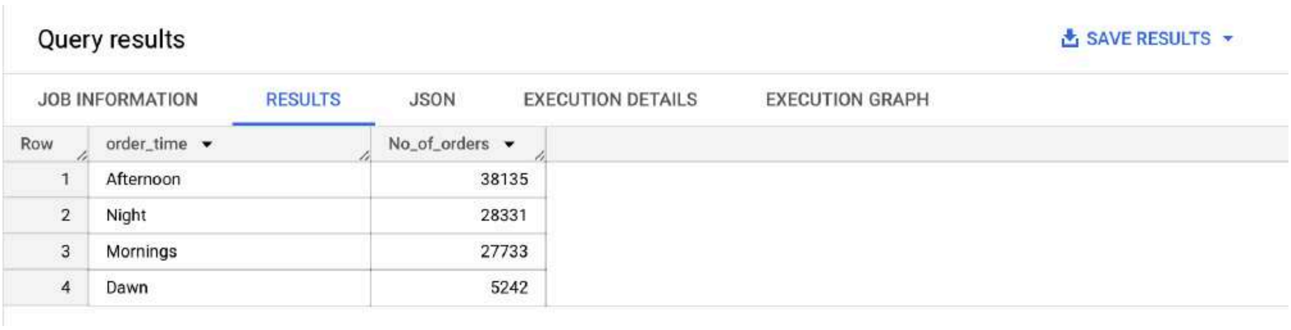
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Order_Year	Order_Month	No_of_orders
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

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

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 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'
else 'Night' end as order_time, count(order_id) as No_of_orders
from `TargetCaseStudy.orders`
group by order_t
order by No_of_orders
```



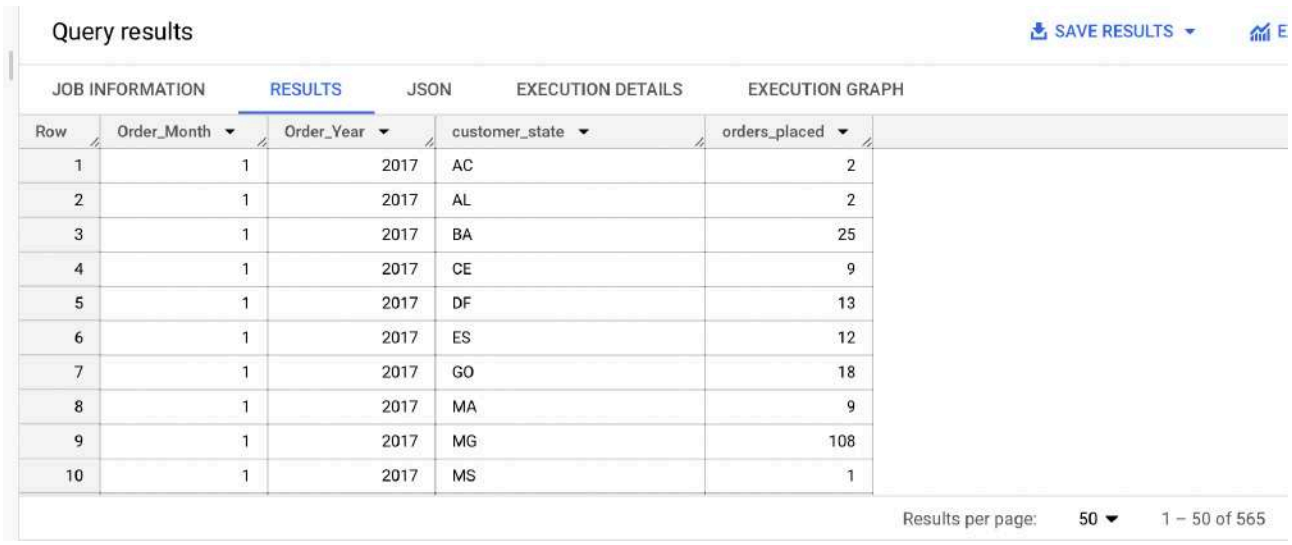
Query results

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

3. Evolution of E-commerce orders in the Brazil region:

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

```
SELECT extract(MONTH from o.order_purchase_timestamp) as Order_Month, extract(Year from
o.order_purchase_timestamp) as Order_Year,
c.customer_state, COUNT(order_id) AS orders_placed
FROM `TargetCaseStudy.orders` o
left join `TargetCaseStudy.customers` c
on o.customer_id=c.customer_id
GROUP BY Order_Month, Order_Year, customer_state
ORDER BY Order_Month, Order_Year, customer_state;
```



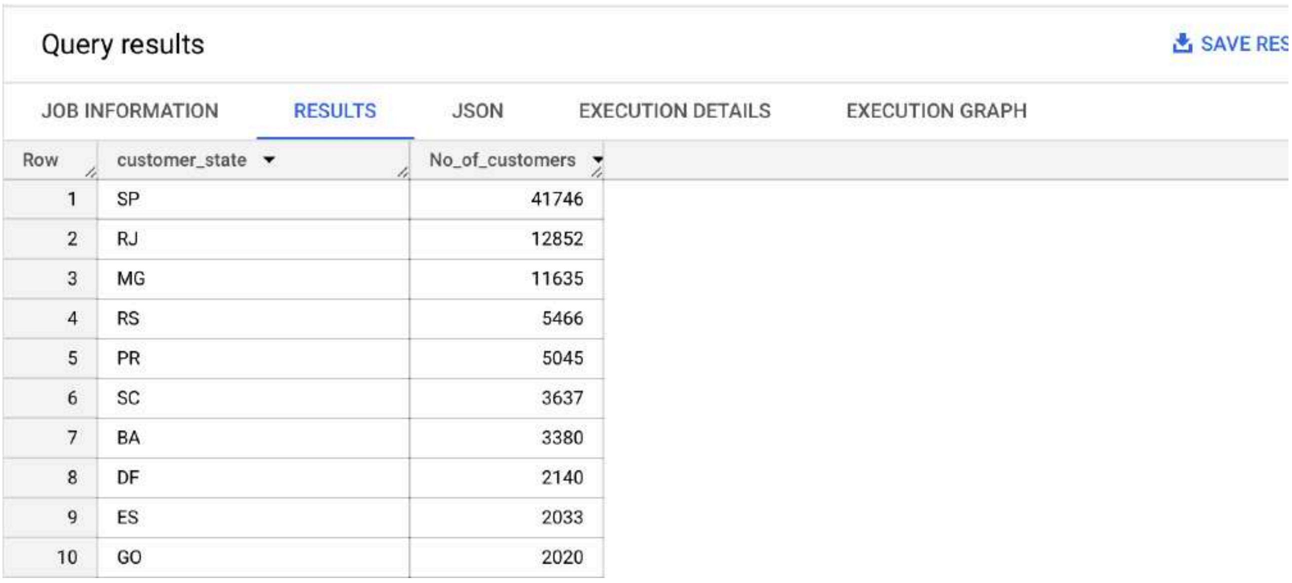
Query results

Row	Order_Month	Order_Year	customer_state	orders_placed
1	1	2017	AC	2
2	1	2017	AL	2
3	1	2017	BA	25
4	1	2017	CE	9
5	1	2017	DF	13
6	1	2017	ES	12
7	1	2017	GO	18
8	1	2017	MA	9
9	1	2017	MG	108
10	1	2017	MS	1

Results per page: 50 1 – 50 of 565

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

```
select customer_state, count(customer_id) as No_of_customers from
`TargetCaseStudy.customers`
group by customer_state
order by No_of_customers desc;
```



Row	customer_state	No_of_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

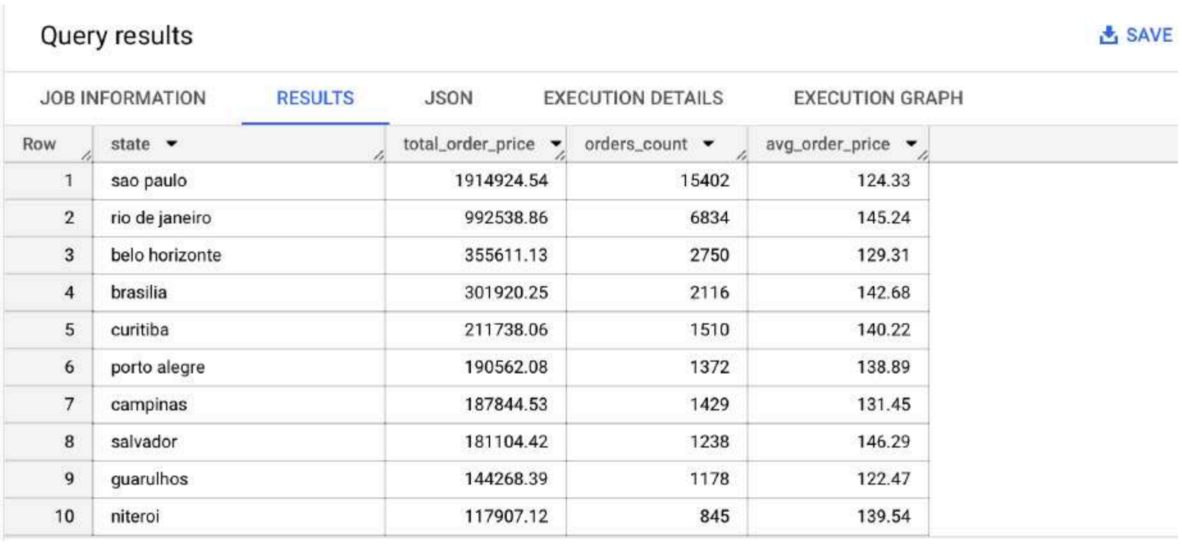
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

4.1 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 FORMAT_DATE('%Y-%m', o.order_purchase_timestamp) AS Year_month,
p.payment_value from `TargetCaseStudy.orders` o
join `TargetCaseStudy.payments` p
on o.order_id=p.order_id
where extract(month from order_purchase_timestamp) >= 1 and
extract(month from order_purchase_timestamp) <= 8
order by payment_value;
```

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

```
with order_details as
(
select c.customer_city as state,
round(sum(oi.price),2) as total_order_price,
count(distinct o.order_id) as orders_count,
from `TargetCaseStudy.orders` o
inner join `TargetCaseStudy.order_items` oi
on o.order_id=oi.order_id
inner join `TargetCaseStudy.customers` c
on o.customer_id=c.customer_id
group by state
)
select
state, total_order_price,
orders_count,
round(total_order_price/orders_count,2) as avg_order_price
from order_details
order by total_order_price desc;
```

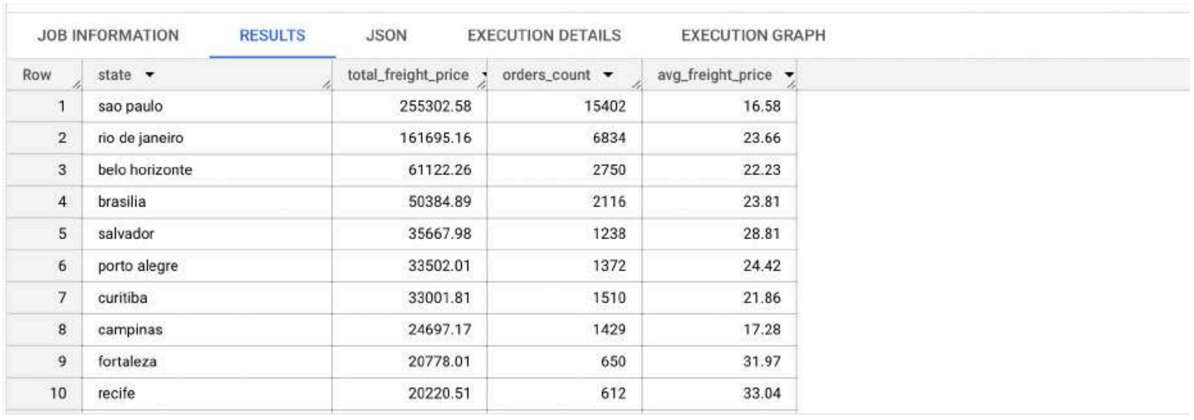


Row	state	total_order_price	orders_count	avg_order_price
1	sao paulo	1914924.54	15402	124.33
2	rio de janeiro	992538.86	6834	145.24
3	belo horizonte	355611.13	2750	129.31
4	brasilia	301920.25	2116	142.68
5	curitiba	211738.06	1510	140.22
6	porto alegre	190562.08	1372	138.89
7	campinas	187844.53	1429	131.45
8	salvador	181104.42	1238	146.29
9	guarulhos	144268.39	1178	122.47
10	niteroi	117907.12	845	139.54

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

```
with order_details as
(
select c.customer_city as state,
round(sum(oi.freight_value),2) as total_freight_price,
count(distinct o.order_id) as orders_count,
from `TargetCaseStudy.orders` o
inner join `TargetCaseStudy.order_items` oi
on o.order_id=oi.order_id
inner join `TargetCaseStudy.customers` c
on o.customer_id=c.customer_id
group by state
)

select
state, total_freight_price,
orders_count,
round(total_freight_price/orders_count,2) as avg_freight_price
from order_details
order by total_freight_price desc;
```



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	state	total_freight_price	orders_count	avg_freight_price
1	sao paulo	255302.58	15402	16.58
2	rio de janeiro	161695.16	6834	23.66
3	belo horizonte	61122.26	2750	22.23
4	brasilia	50384.89	2116	23.81
5	salvador	35667.98	1238	28.81
6	porto alegre	33502.01	1372	24.42
7	curitiba	33001.81	1510	21.86
8	campinas	24697.17	1429	17.28
9	fortaleza	20778.01	650	31.97
10	recife	20220.51	612	33.04

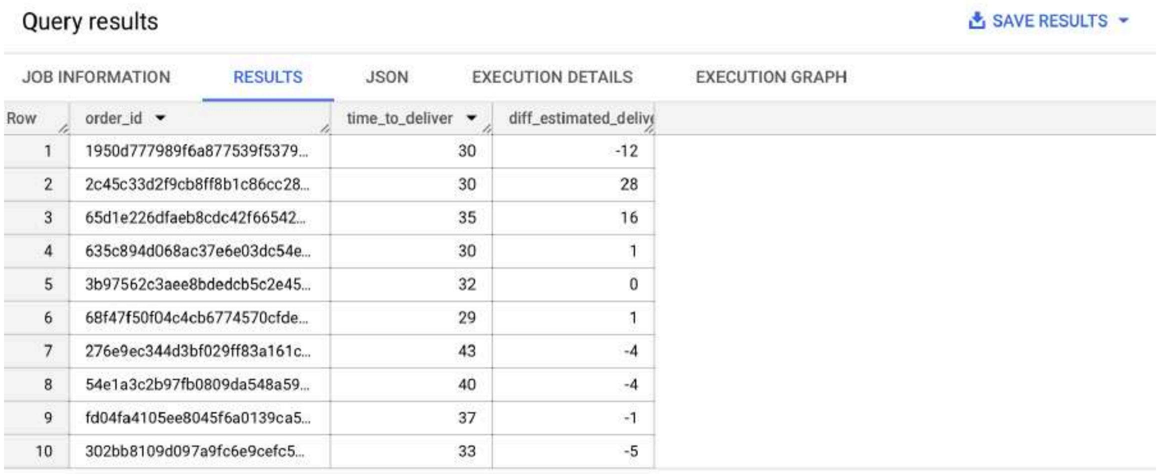
5. Analysis based on sales, freight and delivery time.

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.

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_purchase_timestamp, day) as time_to_deliver,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as
diff_estimated_delivery
FROM
```



Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

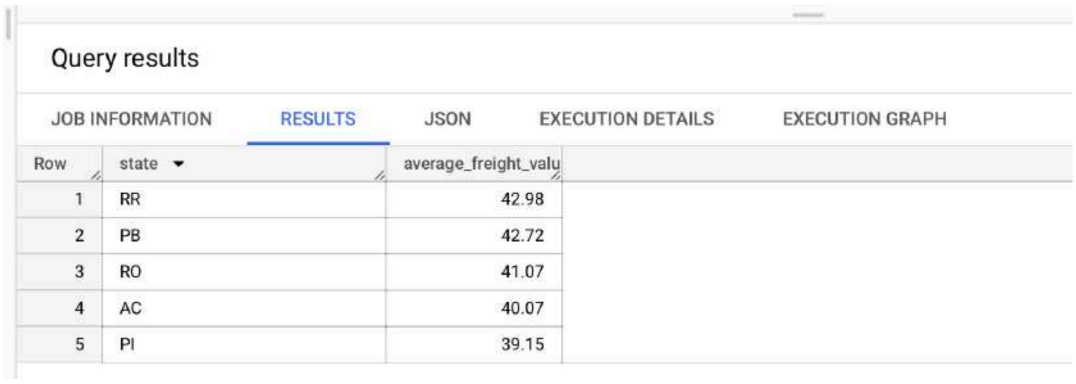
Row	order_id	time_to_deliver	diff_estimated_deliv
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

```
`TargetCaseStudy.orders`;
```

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

- Query for TOP 5 states by highest average freight value.

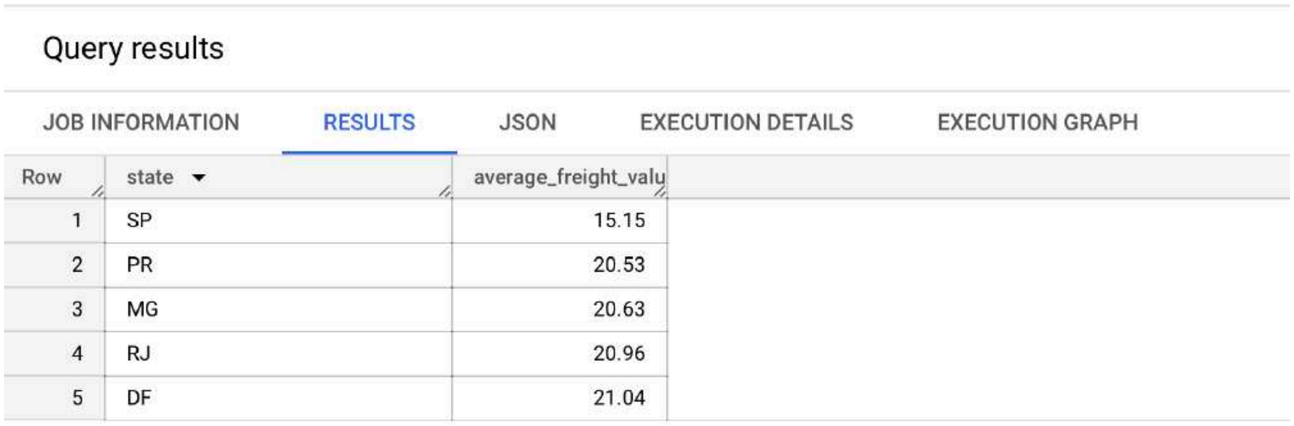
```
SELECT customer_state as state, round(avg(freight_value),2) as average_freight_value
FROM `TargetCaseStudy.orders` o
JOIN `TargetCaseStudy.order_items` oi
ON o.order_id = oi.order_id
JOIN `TargetCaseStudy.customers` c
ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY average_freight_value DESC
LIMIT 5;
```



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

- Query for TOP 5 states by lowest average freight value.

```
SELECT customer_state as state, avg(freight_value) as average_freight_value
FROM `TargetCaseStudy.orders` o
JOIN `TargetCaseStudy.order_items` oi
ON o.order_id = oi.order_id
JOIN `TargetCaseStudy.customers` c
ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY average_freight_value asc
LIMIT 5;
```



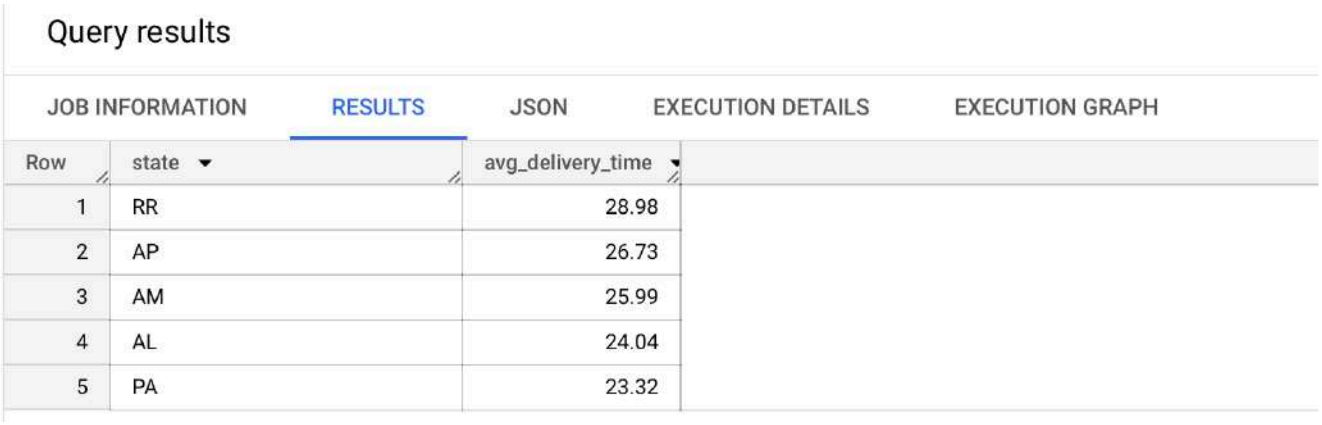
Query results

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

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

- Query for Top 5 states with the highest average delivery time.

```
SELECT c.customer_state as state,
round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)),2) as
avg_delivery_time
FROM `TargetCaseStudy.orders` o
LEFT JOIN `TargetCaseStudy.customers` c
on o.customer_id=c.customer_id
GROUP BY customer_state
ORDER BY avg_delivery_time desc
LIMIT 5;
```

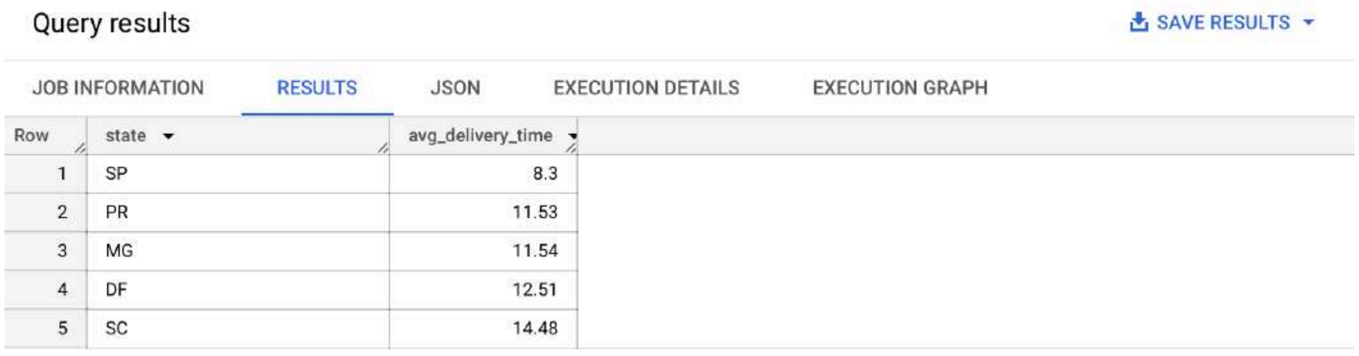


Query results

Row	state	avg_delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

- Query for Top 5 states with the lowest average delivery time.

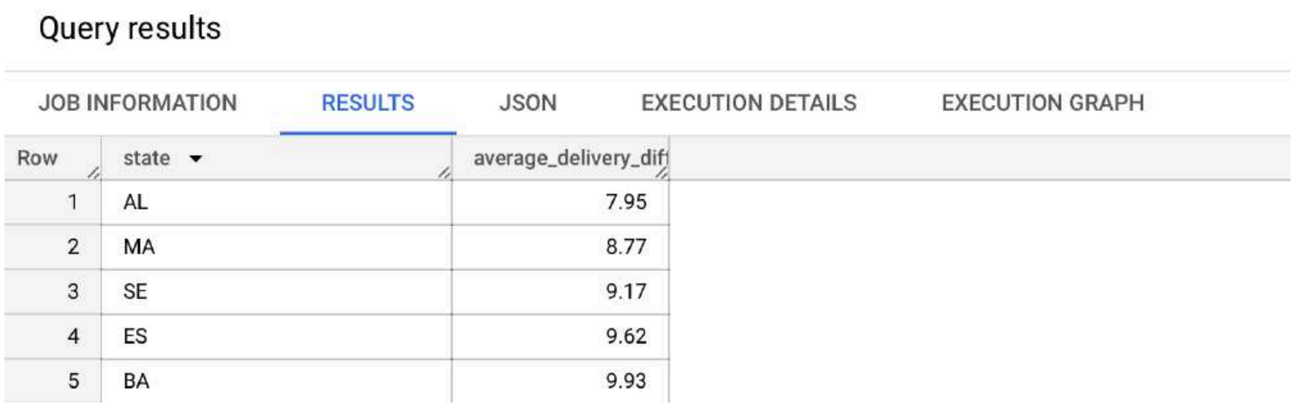
```
SELECT c.customer_state as state,
round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)),2) as
avg_delivery_time
FROM `TargetCaseStudy.orders` o
LEFT JOIN `TargetCaseStudy.customers` c
on o.customer_id=c.customer_id
GROUP BY customer_state
ORDER BY avg_delivery_time
LIMIT 5;
```



Row	state	avg_delivery_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

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

```
SELECT c.customer_state as state,
avg (date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, day)) as
average_delivery_difference
FROM `TargetCaseStudy.orders` o
join `TargetCaseStudy.customers` c
on o.customer_id=c.customer_id
GROUP BY customer_state
ORDER BY average_delivery_difference
LIMIT 5;
```

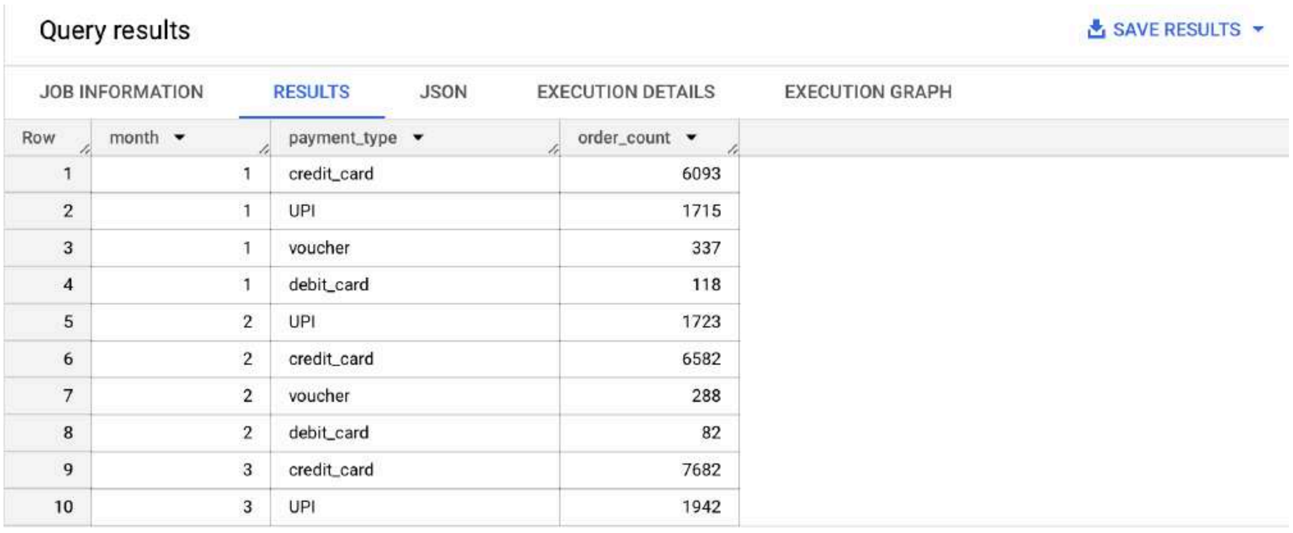


Row	state	average_delivery_dif
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

6. Analysis based on the payments:

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

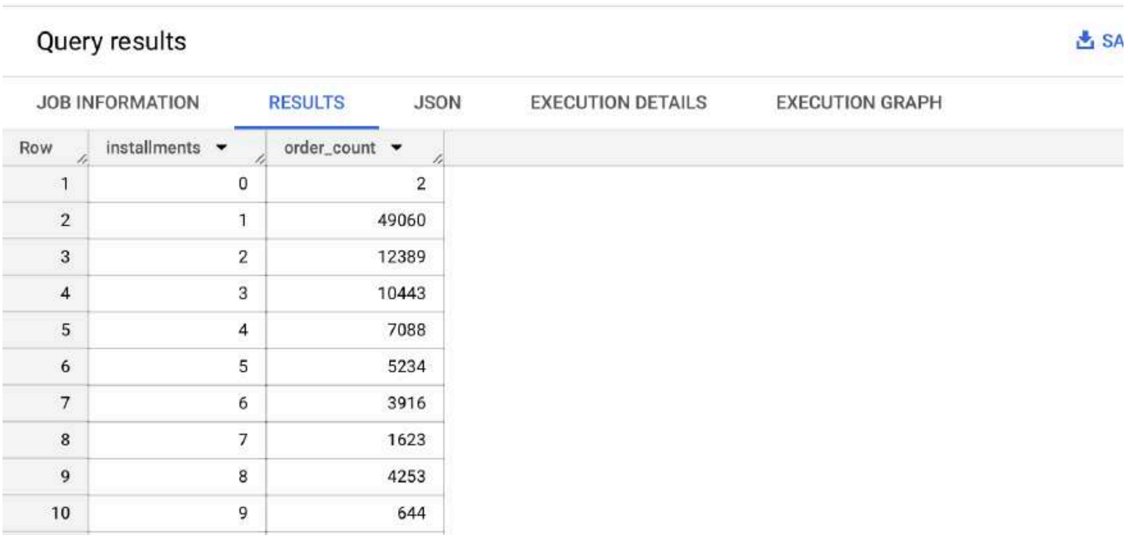
```
SELECT extract(month from o.order_purchase_timestamp) as month,
p.payment_type,
COUNT(DISTINCT o.order_id) as order_count
FROM `TargetCaseStudy.orders` o
JOIN `TargetCaseStudy.payments` p
ON o.order_id = p.order_id
GROUP BY month, payment_type
ORDER BY month;
```



Row	month	payment_type	order_count
1	1	credit_card	6093
2	1	UPI	1715
3	1	voucher	337
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6582
7	2	voucher	288
8	2	debit_card	82
9	3	credit_card	7682
10	3	UPI	1942

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

```
SELECT payment_installments as installments, COUNT(DISTINCT order_id) AS order_count
FROM `TargetCaseStudy.payments` p
GROUP BY payment_installments
ORDER BY payment_installments;
```



Row	installments	order_count
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644