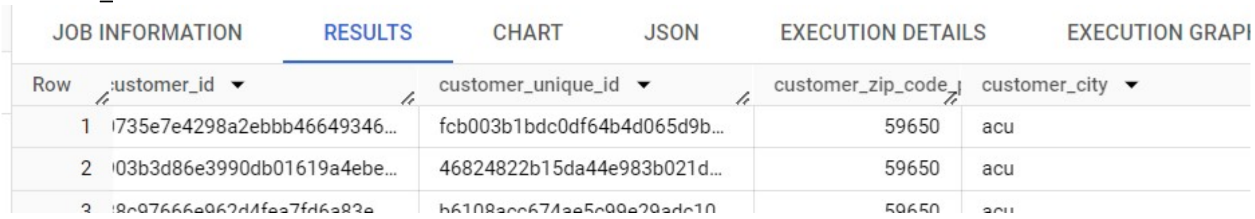
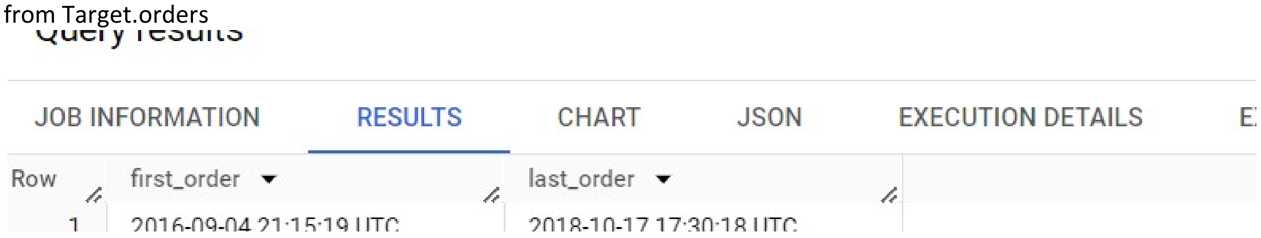
Business Case: Target SQL

Problem Statement:
Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.
What does 'good' look like?
1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
1.Data type of all columns in the "customers" table.
ANS: customer_id – VARCHAR
customer_unique_id - VARCHAR
customer_zip_code_prefix – INT
customer_city – CHAR
customer_state – CHAR


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

Row	:ustomer_id	customer_unique_id	customer_zip_code_	customer_city
1	735e7e4298a2ebbb46649346...	fcb003b1bdc0df64b4d065d9b...	59650	acu
2	03b3d86e3990db01619a4ebe...	46824822b15da44e983b021d...	59650	acu
3	8c97666e962d4fea7fd6a83e	b6108acc674ae5c99e29adc10	59650	acu

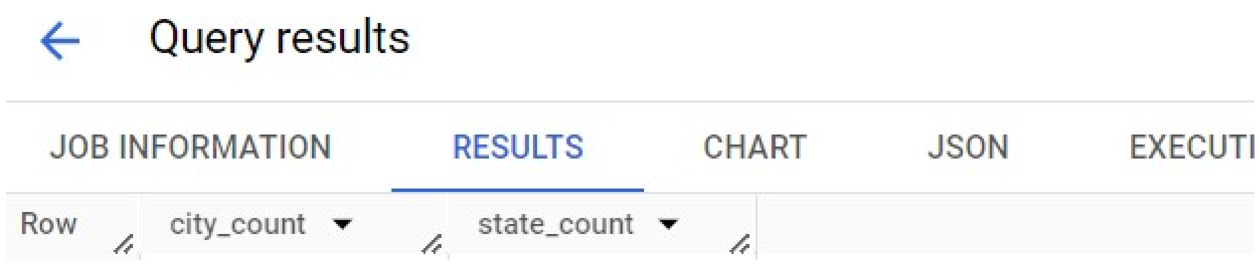
2. Get the time range between which the orders were placed.
ANS:
Select
min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order
from Target.orders


Query results

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

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

3. Count the Cities & States of customers who ordered during the given period.
ANS:
```
Select count(distinct geolocation_city) as city_count, count(distinct geolocation_state) as state_count
from Target.geolocation GT join Target.customers CT on
GT.geolocation_zip_code_prefix=CT.customer_zip_code_prefix
join Target.orders OT on CT.customer_id=OT.customer_id
where order_purchase_timestamp between '2016-09-04 21:15:19 UTC' and '2018-10-17 17:30:18 UTC';
```


Query results

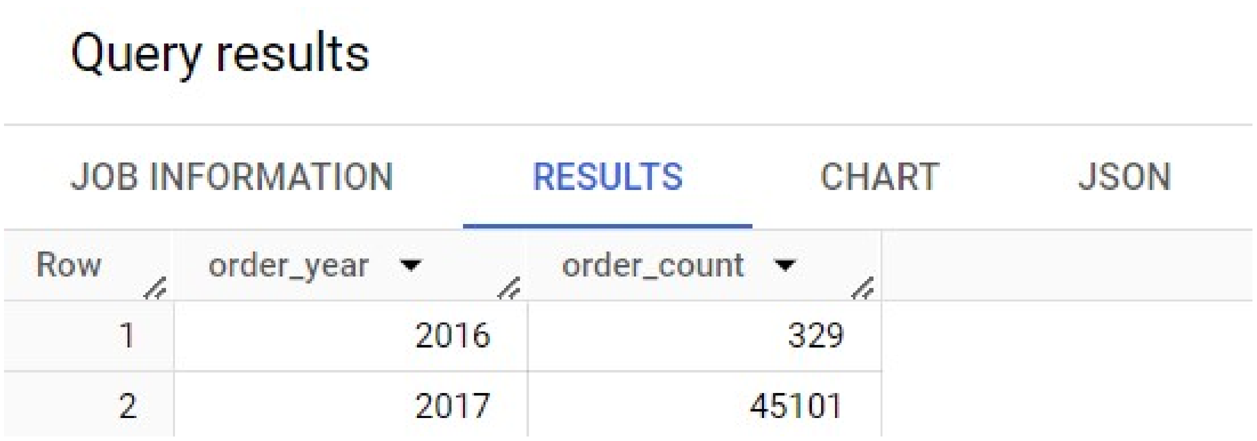
JOB INFORMATION | RESULTS | CHART | JSON | EXECUTI

Row	city_count	state_count

2)In-depth Exploration:
1.Is there a growing trend in the no. of orders placed over the past years?
ANS:

```
select
extract(year from order_purchase_timestamp) as order_year,
count(*) as order_count
from
target.orders
group by
extract(year from order_purchase_timestamp)
order by order_year;
```

Insights: The growing trend in the number of orders over the years is increasing significantly.

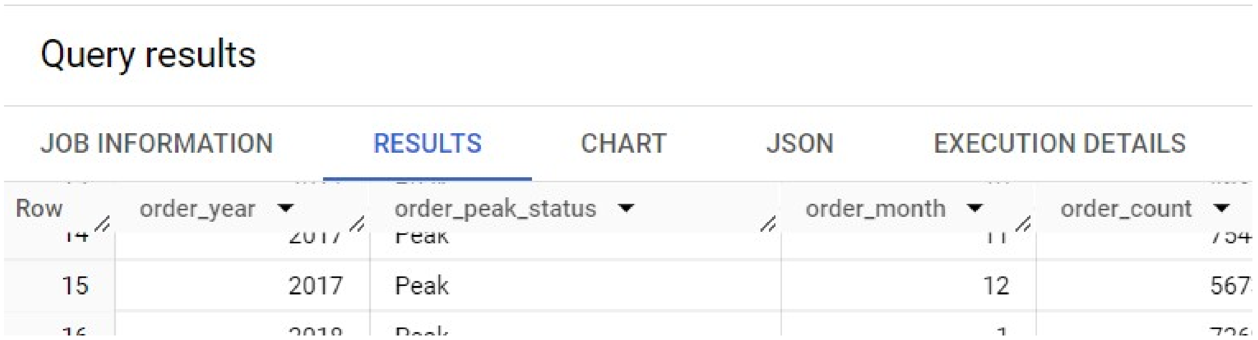


Row	order_year	order_count
1	2016	329
2	2017	45101

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

```
with monthlyordercounts as (
select
extract(year from order_purchase_timestamp) as order_year,
extract(month from order_purchase_timestamp) as order_month,
count(*) as order_count
from
target.orders
group by
extract(year from order_purchase_timestamp),
extract(month from order_purchase_timestamp)
)
select
order_year,
case
when order_count > (select avg(order_count) from monthlyordercounts) then 'peak'
else 'non-peak'
end as order_peak_status,
order_month,
order_count
from
monthlyordercounts
```

```
order by
order_year, order_month;
```



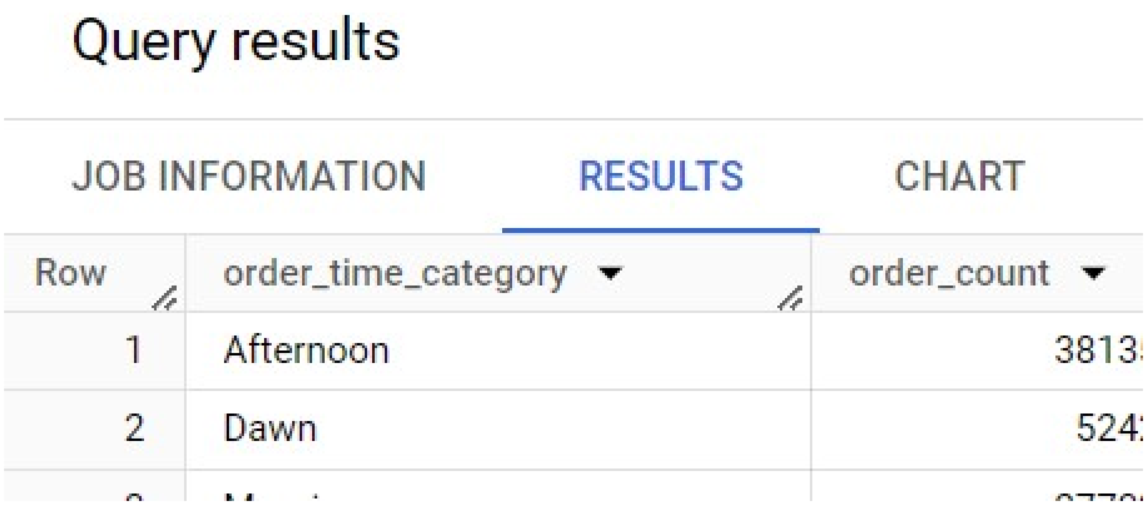
Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	order_year	order_peak_status	order_month	order_count
14	2017	Peak	11	754
15	2017	Peak	12	567
16	2018	Peak	1	726

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 : Nightbeing placed?
ANS

```
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_category,
count(*) as order_count
from Target.orders
group by order_time_category
order by order_time_category;
```



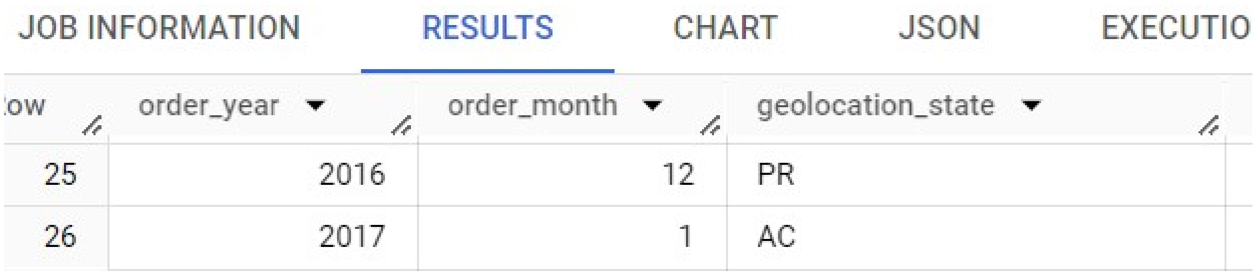
Query results

JOB INFORMATION | RESULTS | CHART

Row	order_time_category	order_count
1	Afternoon	3813
2	Dawn	524

3)Evolution of E-commerce orders in the Brazil region:
1.Get the month on month no. of orders placed in each state.
ANS:

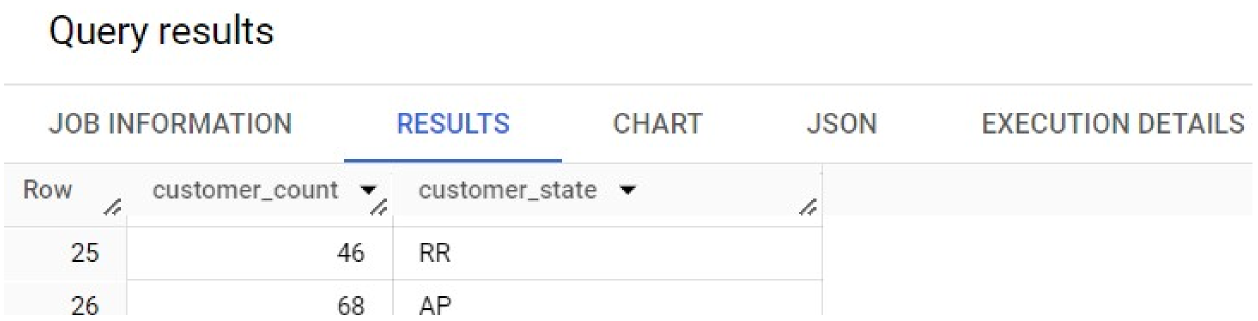
```
Select
extract(Year from order_purchase_timestamp)as order_year,
extract(Month from order_purchase_timestamp)as order_month,
geolocation_state,
count(*) as order_count
from Target.geolocation GT join Target.customers CT
on GT.geolocation_zip_code_prefix=CT.customer_zip_code_prefix
join Target.orders OT on CT.customer_id=OT.customer_id
Group by extract(Year from order_purchase_timestamp),
extract(Month from order_purchase_timestamp),
geolocation_state
order by order_year,order_month,geolocation_state;
```



ow	order_year	order_month	geolocation_state
25	2016	12	PR
26	2017	1	AC

2. How are the customers distributed across all the states?
ANS:

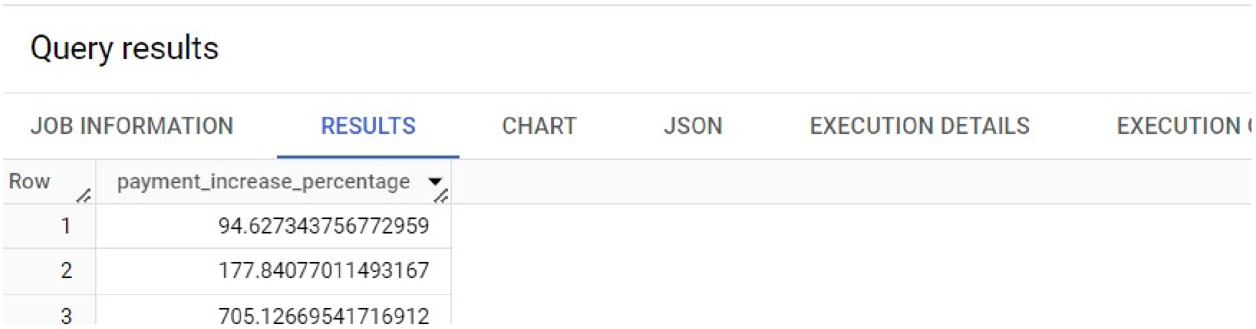
```
Select count(distinct customer_id) as customer_count,
customer_state from Target.customers
group by customer_state;
```



Row	customer_count	customer_state
25	46	RR
26	68	AP

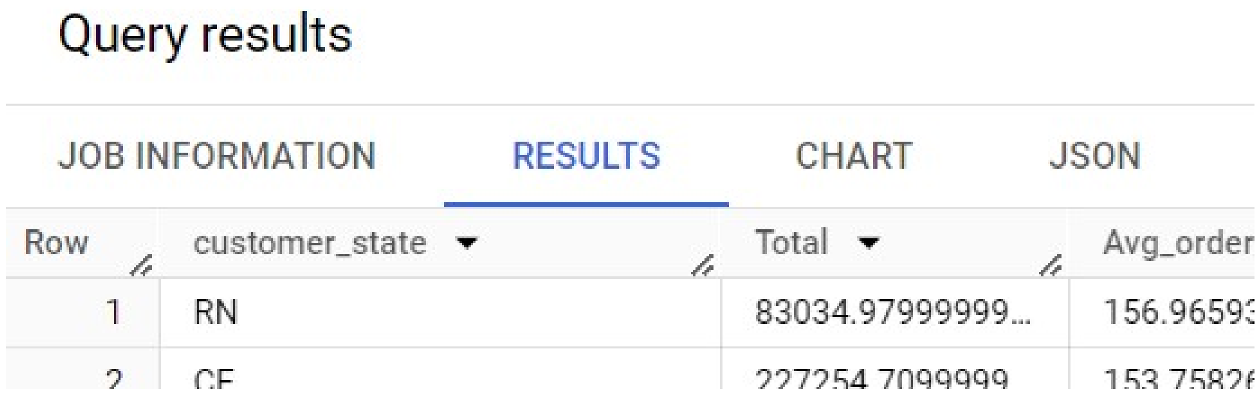
4)Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
1.Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

ANS
```
with orderpayment as (
select
extract(year from o.order_purchase_timestamp) as order_year,
extract(month from o.order_purchase_timestamp) as order_month,
sum(p.payment_value) as total_payment
from target.orders o join
target.payments p on p.order_id = o.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
extract(year from o.order_purchase_timestamp),
extract(month from o.order_purchase_timestamp)
)
select
((op_2018.total_payment - op_2017.total_payment) / nullif(op_2017.total_payment, 0)) * 100 as
payment_increase_percentage
from
orderpayment op_2017
join
orderpayment op_2018 on op_2017.order_month = op_2018.order_month
where
op_2017.order_year = 2017
and op_2018.order_year = 2018;
```



Row	payment_increase_percentage
1	94.627343756772959
2	177.84077011493167
3	705.12669541716912

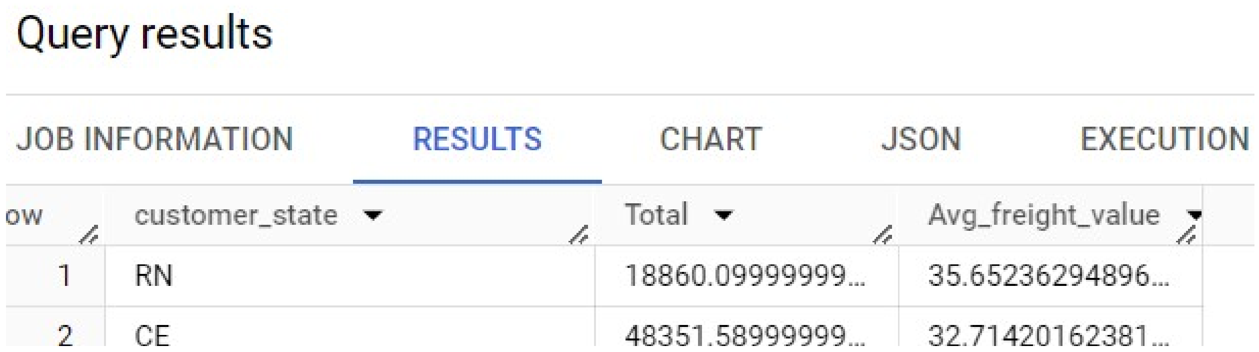
2. Calculate the Total & Average value of order price for each state.
ANS:
```
Select c.customer_state,sum(oi.price) Total,Avg(oi.price)Avg_order_price
from Target.customers c
left join Target.orders o on c.customer_id=o.customer_id
left join Target.order_items oi on o.order_id=oi.order_id
group by c.customer_state
```



Row	customer_state	Total	Avg_order
1	RN	83034.97999999...	156.9659
2	CE	227254.7099999	153.7582

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

```
Select  c.customer_state,sum(oi.freight_value) Total,Avg(oi.freight_value)Avg_freight_value
from Target.customers c
left join Target.orders o on c.customer_id=o.customer_id
left join Target.order_items oi on o.order_id=oi.order_id
group by c.customer_state;
```

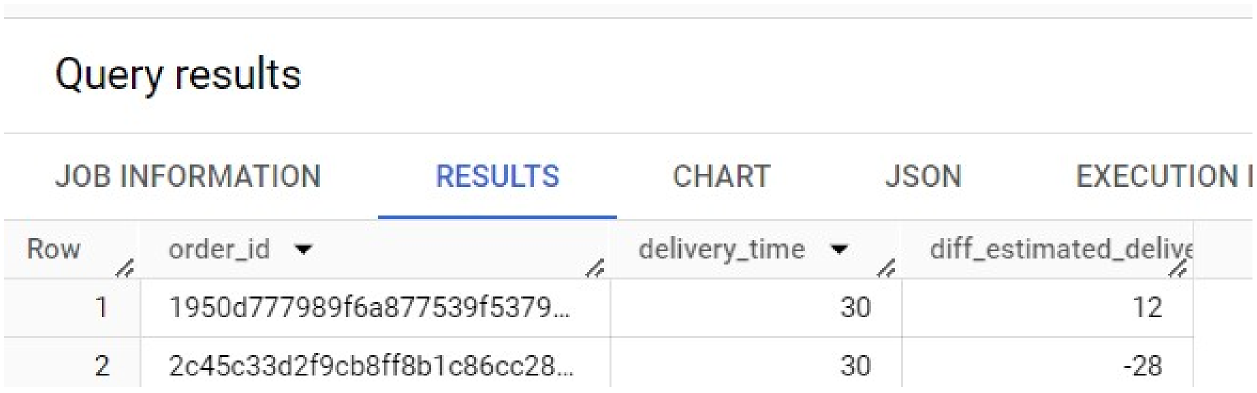


ow	customer_state	Total	Avg_freight_value
1	RN	18860.09999999...	35.65236294896...
2	CE	48351.58999999...	32.71420162381...

5)Analysis based on sales, freight and delivery time.
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_delivered_customer_date - order_estimated_delivery_date

ANS:

```
select
order_id,
timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day) as delivery_time,
timestamp_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as diff_estimated_delivery
from Target.orders ;
```

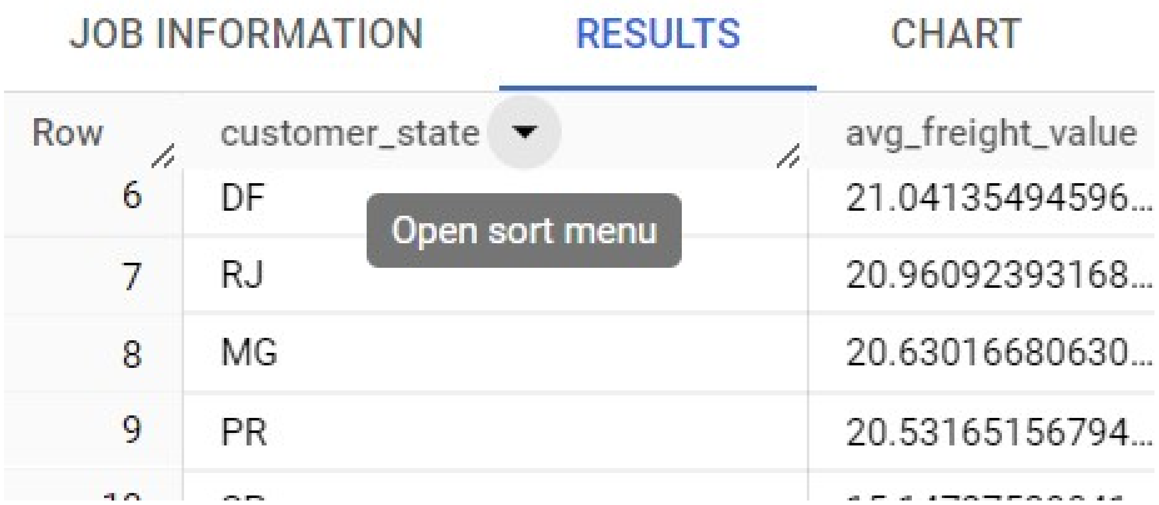

Query results

JOB INFORMATION RESULTS CHART JSON EXECUTION I

Row	order_id	delivery_time	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	-28

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

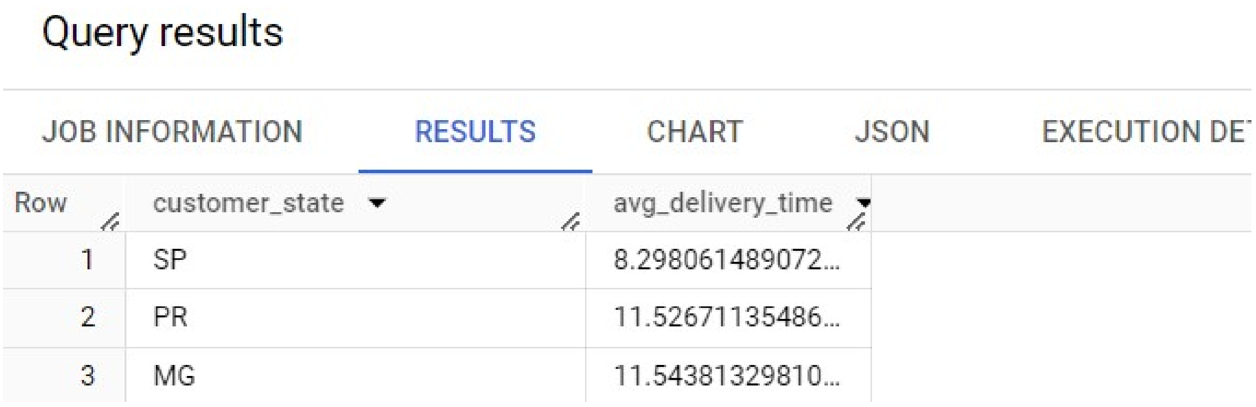
```
with StateFreight as (
select
c.customer_state,
avg(oi.freight_value) AS avg_freight_value,
row_number() over (order by  avg(oi.freight_value) desc) as high_rank,
row_number() over (order by avg(oi.freight_value)) as low_rank
from Target.customers c
left join Target.orders o on c.customer_id = o.customer_id
left join Target.order_items oi ON o.order_id = oi.order_id
group by c.customer_state
)
select
customer_state,
avg_freight_value
from StateFreight
where high_rank <= 5
union all
select
customer_state,
avg_freight_value
from StateFreight
where low_rank <= 5
order by avg_freight_value desc;
```


JOB INFORMATION RESULTS CHART

Row	customer_state	avg_freight_value
6	DF	21.04135494596...
7	RJ	20.96092393168...
8	MG	20.63016680630...
9	PR	20.53165156794...

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

```
with DeliveryTime as (
select c.customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, DAY)) as avg_delivery_time,
row_number() over (order by avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, DAY))
DESC) as high_rank,
row_number() over (order BY avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, DAY)))
as low_rank
from Target.orders o
join Target.customers c ON o.customer_id = c.customer_id
group by c.customer_state
)
select
customer_state,
avg_delivery_time
from DeliveryTime
where high_rank <= 5
union all
select
customer_state,
avg_delivery_time
from DeliveryTime
where low_rank <= 5
order by avg_delivery_time;
```

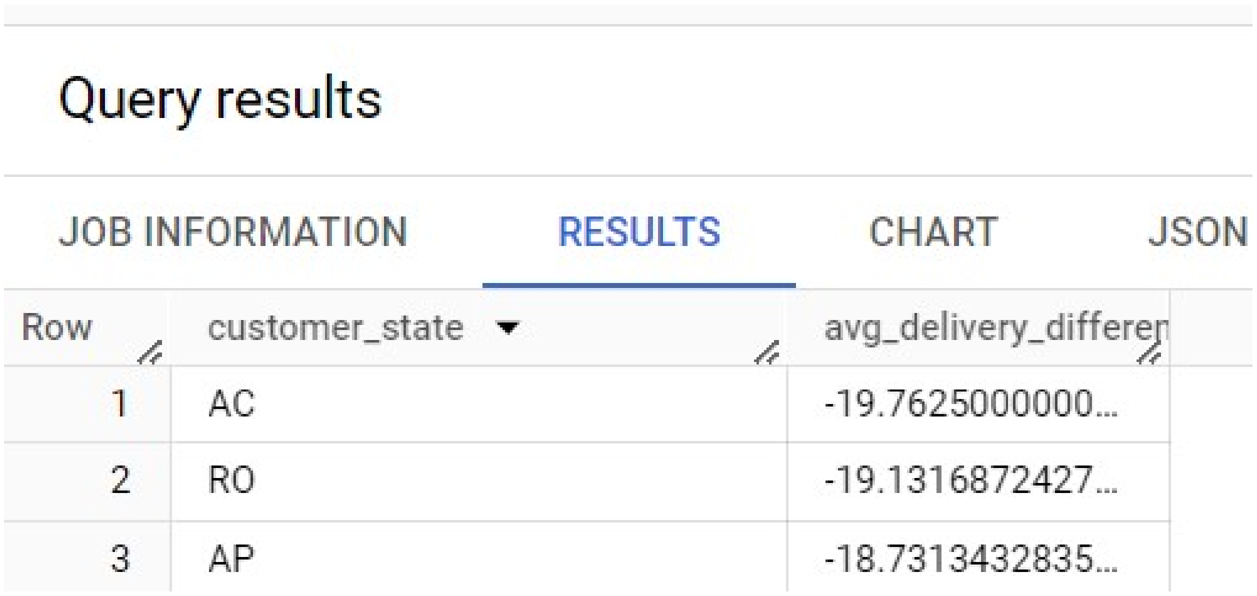

Query results

JOB INFORMATION RESULTS CHART JSON EXECUTION DE

Row	customer_state	avg_delivery_time
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...

4. 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.
ANS:

```
with deliverytime as (
select
c.customer_state,
avg(date_diff(o.order_delivered_customer_date, o.order_estimated_delivery_date, day)) as
avg_delivery_difference
from
target.orders o
join
target.customers c on o.customer_id = c.customer_id
where
o.order_delivered_customer_date is not null
and o.order_estimated_delivery_date is not null
group by
c.customer_state
)
select
customer_state,
avg_delivery_difference
from
deliverytime
order by
avg_delivery_difference asc
limit
5;
```

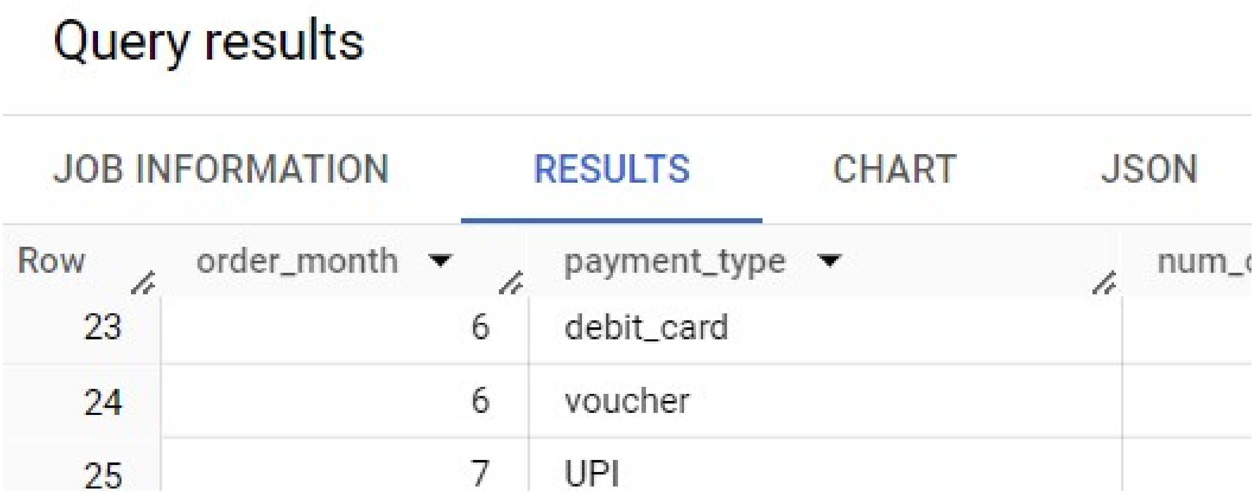

Query results

JOB INFORMATION RESULTS CHART JSON

Row	customer_state	avg_delivery_differen
1	AC	-19.7625000000...
2	RO	-19.1316872427...
3	AP	-18.7313432835...

6)Analysis based on the payments:
1.Find the month on month no. of orders placed using different payment types.
ANS:

```
select
extract(month from o.order_purchase_timestamp) as order_month,
p.payment_type,
count(o.order_id) as num_orders
from
target.orders o
join
target.payments p on o.order_id = p.order_id
group by
extract(month from o.order_purchase_timestamp),
p.payment_type
order by
order_month,
p.payment_type;
```



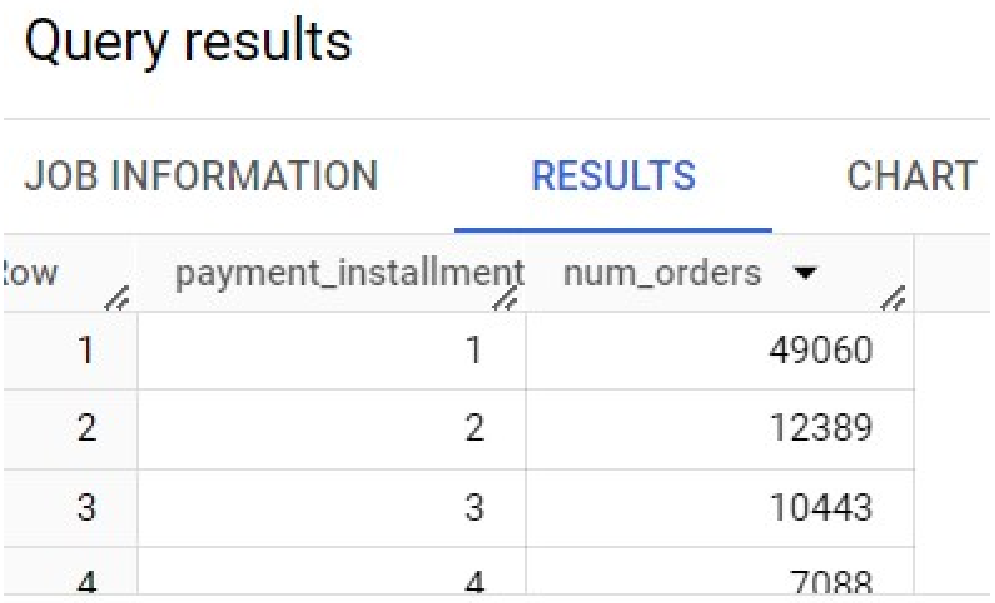
Query results

JOB INFORMATION RESULTS CHART JSON

Row	order_month	payment_type	num_
23	6	debit_card	
24	6	voucher	
25	7	UPI	

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

```
SELECT
p.payment_installments,
COUNT(DISTINCT o.order_id) AS num_orders
FROM
Target.orders o
JOIN
Target.payments p ON o.order_id = p.order_id
WHERE
p.payment_installments > 0
GROUP BY
p.payment_installments
ORDER BY
p.payment_installments;
```



Query results

JOB INFORMATION RESULTS CHART

ow	payment_installment	num_orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088