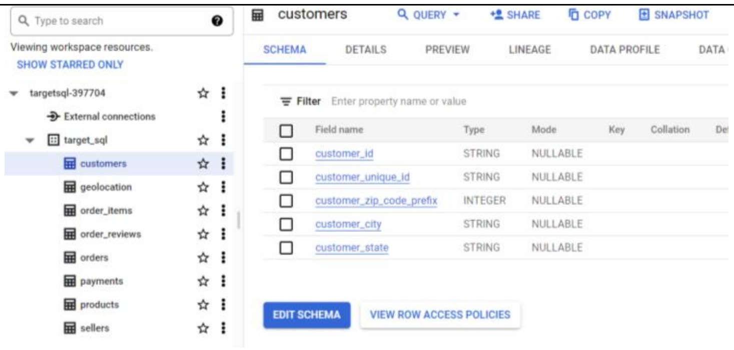
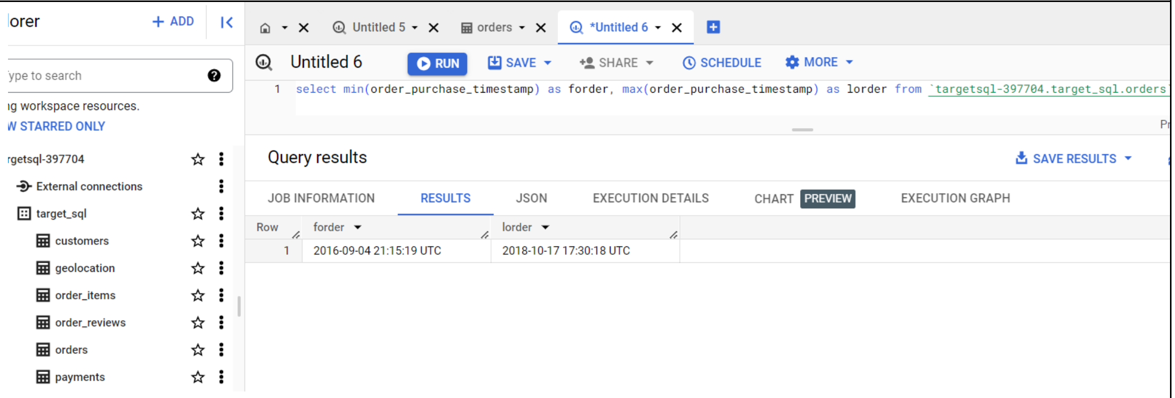
Name :- Harshal Fulzele harshalf786@gmail.com

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

1.1	Data type of all columns in the "customers" table.
	
insights	In BigQuery we can find data type of the column directly by clicking on schema of the table like here we find data type of customer table column e.g. customer_id is of data type string

customers

Field name	Type	Mode	Key	Collation	De
customer_id	STRING	NULLABLE			
customer_unique_id	STRING	NULLABLE			
customer_zip_code_prefix	INTEGER	NULLABLE			
customer_city	STRING	NULLABLE			
customer_state	STRING	NULLABLE			

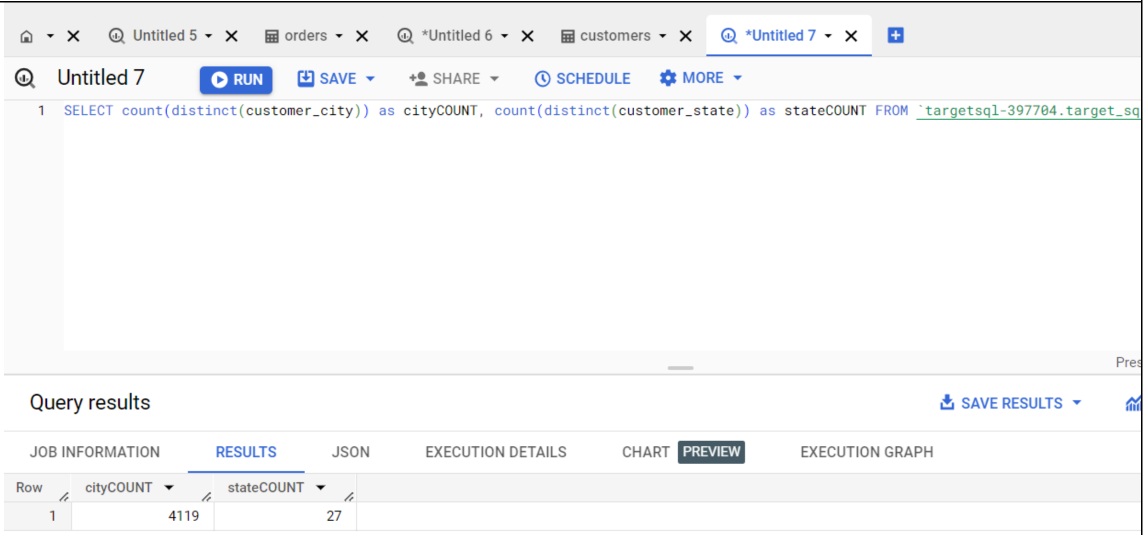
1.2	Get the time range between which the orders were placed
Code+ result	
Insights	Time range between which the order were placed I found that using min and max first of all we need to find where can we get the order timestamp in this we will got in order purchase time so we got time range and I stored in forder and lorder column

```
select min(order_purchase_timestamp) as forder, max(order_purchase_timestamp) as lorder from `targetsql-397704.target_sql.orders
```

Query results

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

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



```
SELECT count(distinct(customer_city)) as cityCOUNT, count(distinct(customer_state)) as stateCOUNT FROM `targetsql-397704.target_sq
```

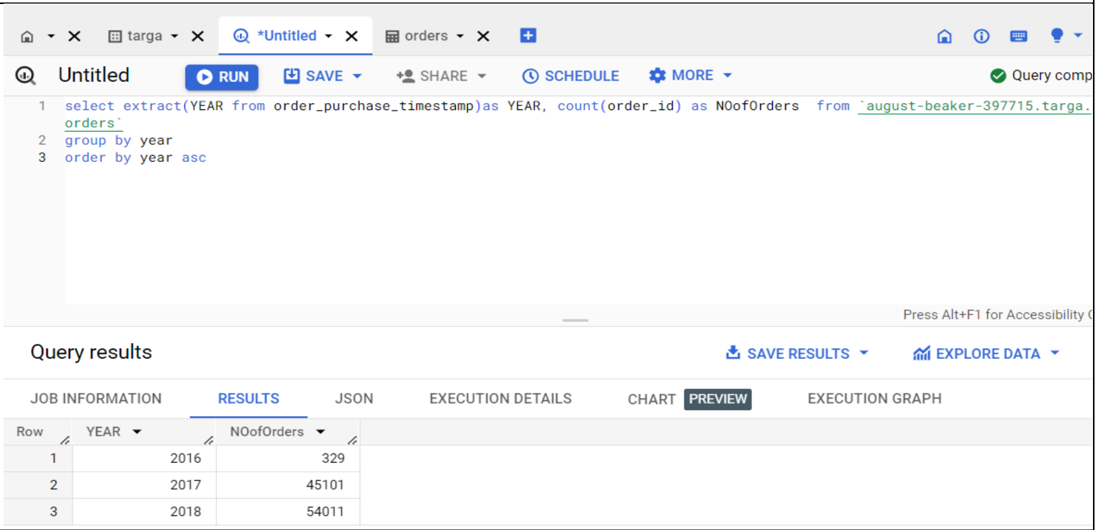
Row	cityCOUNT	stateCOUNT
1	4119	27

We have orders from 4119 cities which we got using unique and we have 27 unique states which we found using the count function.

2. In-depth Exploration:

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

Query + Results

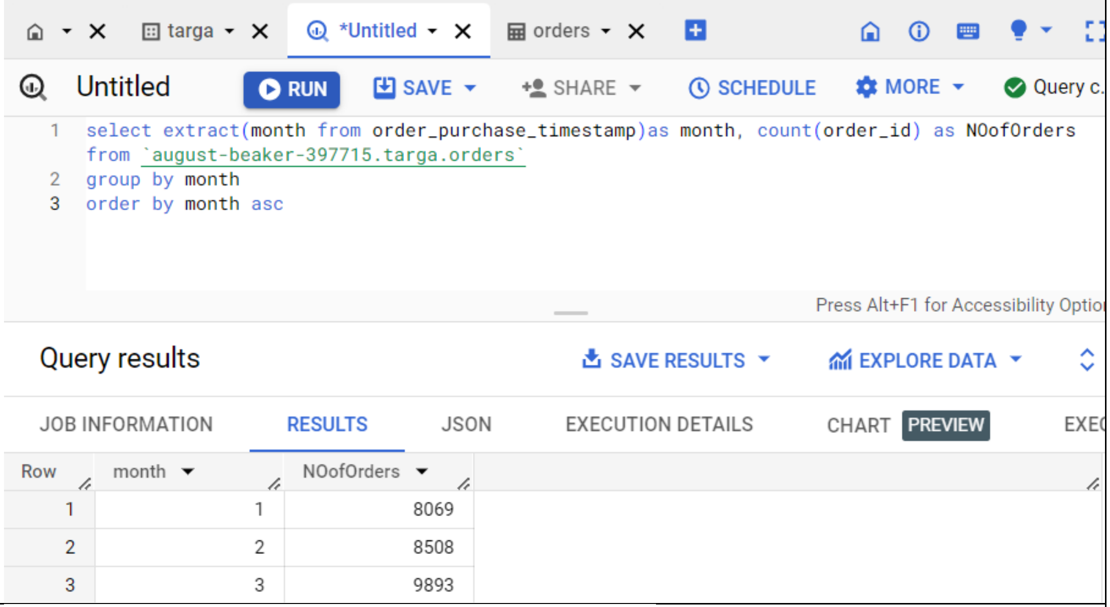
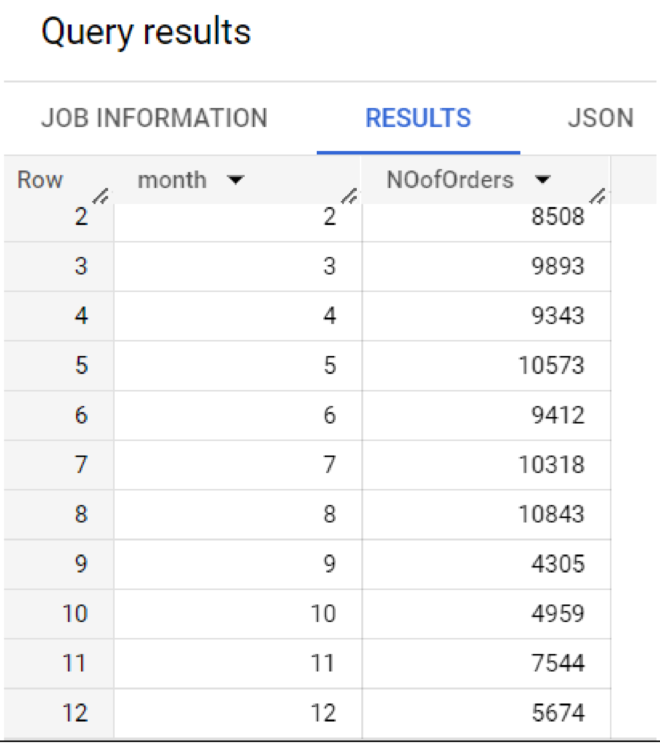


```
select extract(YEAR from order_purchase_timestamp)as YEAR, count(order_id) as NOofOrders  from `august-beaker-397715.targa.
orders`
group by year
order by year asc
```

Row	YEAR	NOofOrders
1	2016	329
2	2017	45101
3	2018	54011

Insights

We can see that there is growing trend from year 2016 to year 2018 initially only 329 orders where there but then there is growth of 54011 orders which is very much we here used the extract function to extract the year and count function to count the no od orders placed in all we used group by clause on basis of year and order by year in ascending order

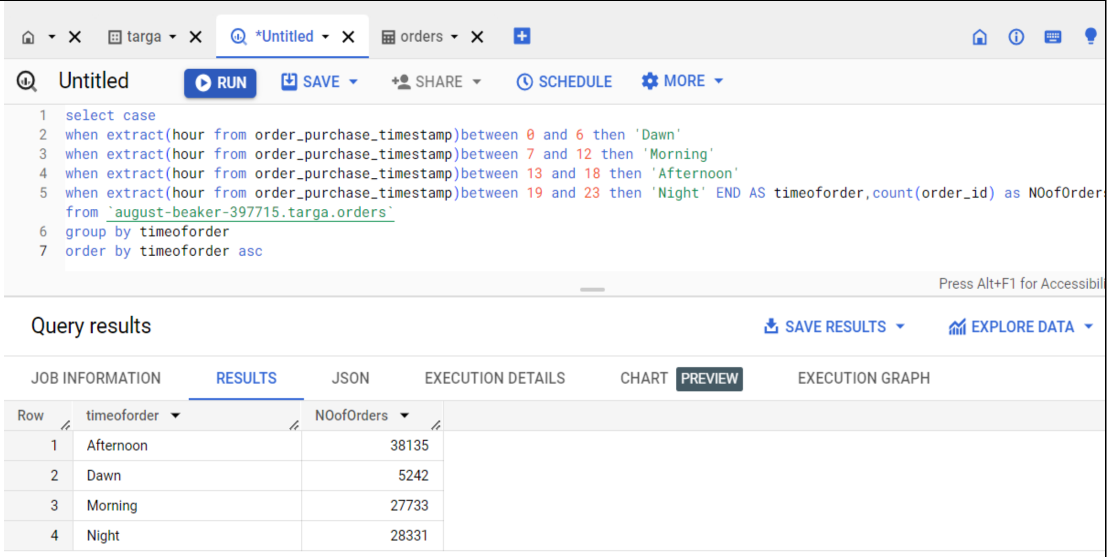
2.2	Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
Query	
Result	
Insights	Here we can see that on month 5 7 and 8 there is monthly trend highest that is month may july and august and lowest in month of 9 that is September we found that using extract function that is extract month so here we can see 12 months and orders accordingly

```
select extract(month from order_purchase_timestamp)as month, count(order_id) as NOofOrders
from `august-beaker-397715.targa.orders`
group by month
order by month asc
```

Row	month	NOofOrders
1	1	8069
2	2	8508
3	3	9893

Row	month	NOofOrders
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

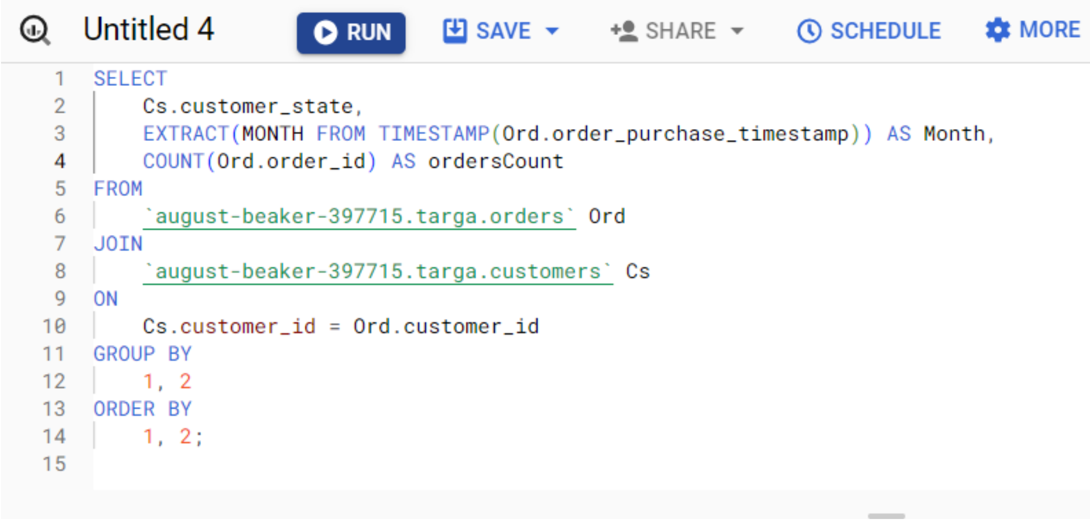
<table>
<tr><td>2.3</td><td>During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night
</td></tr>
<tr><td>Query</td><td>

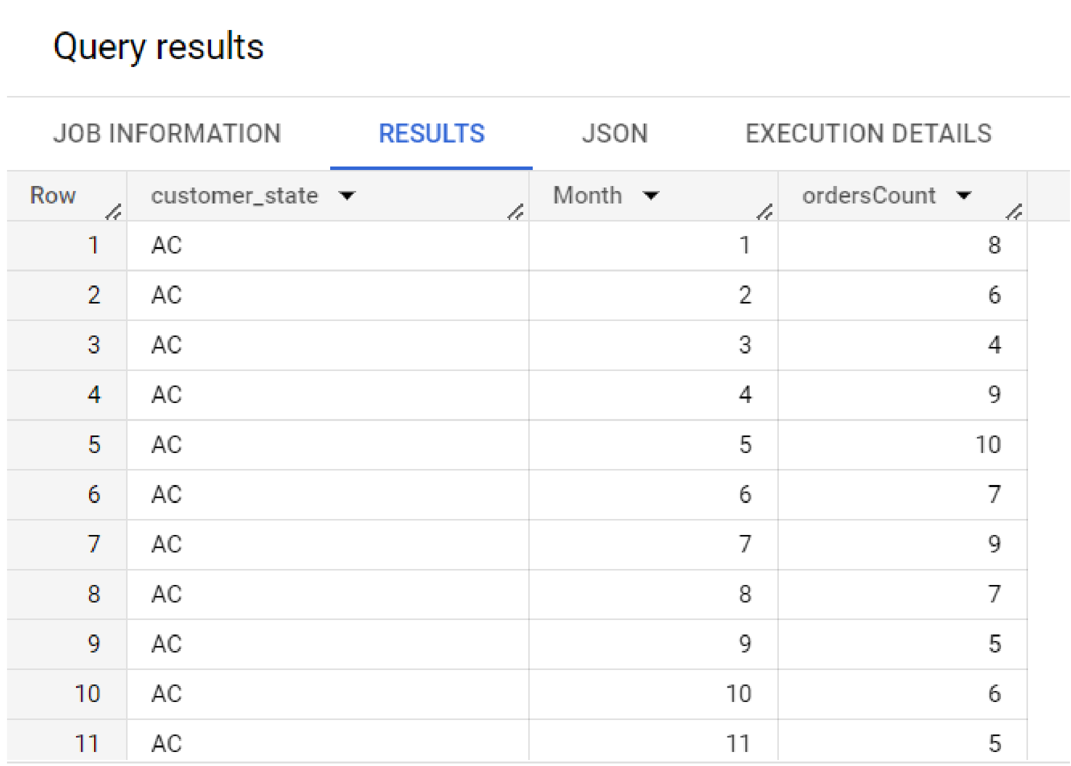
```
select case
when extract(hour from order_purchase_timestamp)between 0 and 6 then 'Dawn'
when extract(hour from order_purchase_timestamp)between 7 and 12 then 'Morning'
when extract(hour from order_purchase_timestamp)between 13 and 18 then 'Afternoon'
when extract(hour from order_purchase_timestamp)between 19 and 23 then 'Night' END AS timeoforder,count(order_id) as NOofOrder
from `august-beaker-397715.targa.orders`
group by timeoforder
order by timeoforder asc
```

<table>
<tr><th>Row</th><th>timeoforder</th><th>NOofOrders</th></tr>
<tr><td>1</td><td>Afternoon</td><td>38135</td></tr>
<tr><td>2</td><td>Dawn</td><td>5242</td></tr>
<tr><td>3</td><td>Morning</td><td>27733</td></tr>
<tr><td>4</td><td>Night</td><td>28331</td></tr>
</table>
</td></tr>
<tr><td>Insights</td><td>In the afternoon time most Brazilians customers place their order and number is 38135</td></tr>
<tr><td></td><td></td></tr>
</table>

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

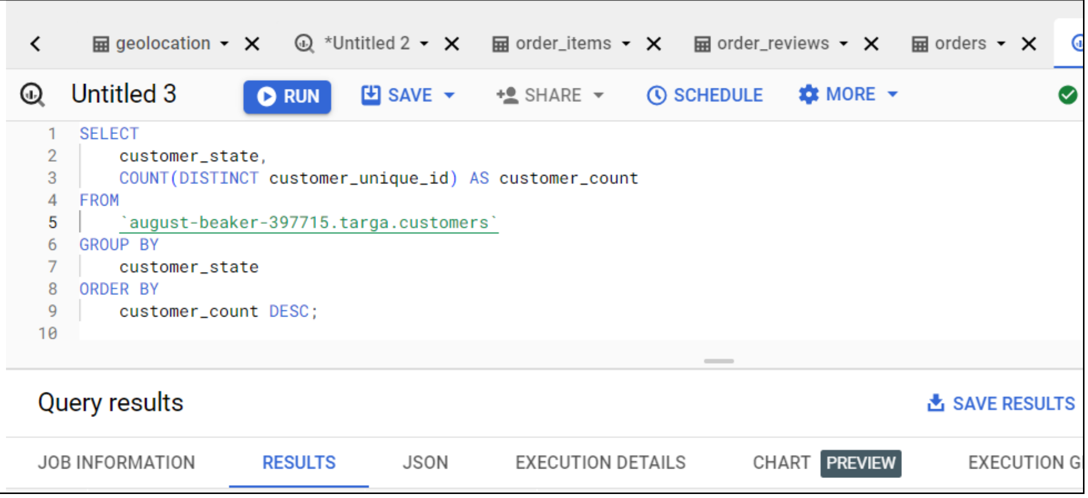
<table>
<tr><td>3.1</td><td>Get the month on month no. of orders placed in each state.</td></tr>
<tr><td></td><td>

```
SELECT
    Cs.customer_state,
    EXTRACT(MONTH FROM TIMESTAMP(Ord.order_purchase_timestamp)) AS Month,
    COUNT(Ord.order_id) AS ordersCount
FROM
    `august-beaker-397715.targa.orders` Ord
JOIN
    `august-beaker-397715.targa.customers` Cs
ON
    Cs.customer_id = Ord.customer_id
GROUP BY
    1, 2
ORDER BY
    1, 2;
```
</td></tr>
</table>

	
	Insight :- Here we count the no. of orders on the basis of the month for each state

Query results

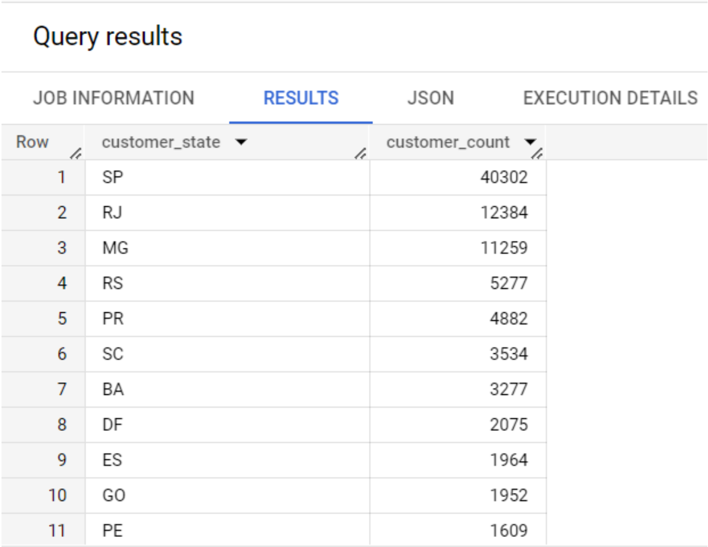
Row	customer_state	Month	ordersCount
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5

3.2	How are the customers distributed across all the states?
	

```sql
SELECT
    customer_state,
    COUNT(DISTINCT customer_unique_id) AS customer_count
FROM
    `august-beaker-397715.targa.customers`
GROUP BY
    customer_state
ORDER BY
    customer_count DESC;

```

Query results



Query results

Row	customer_state	customer_count
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952
11	PE	1609

Insight :- we found that how are customers spread all over the state we found that sp state has more customers for target in brazil.

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

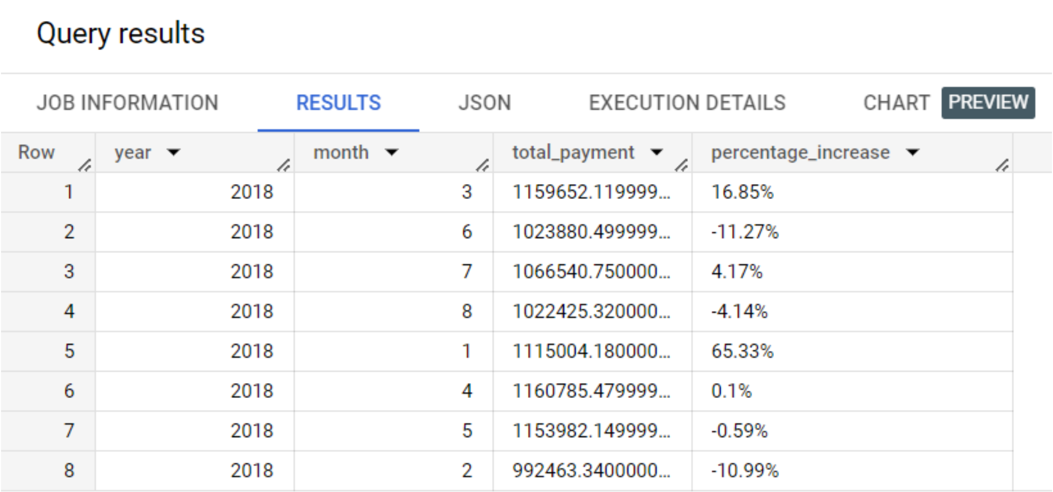
4.1

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.

Query

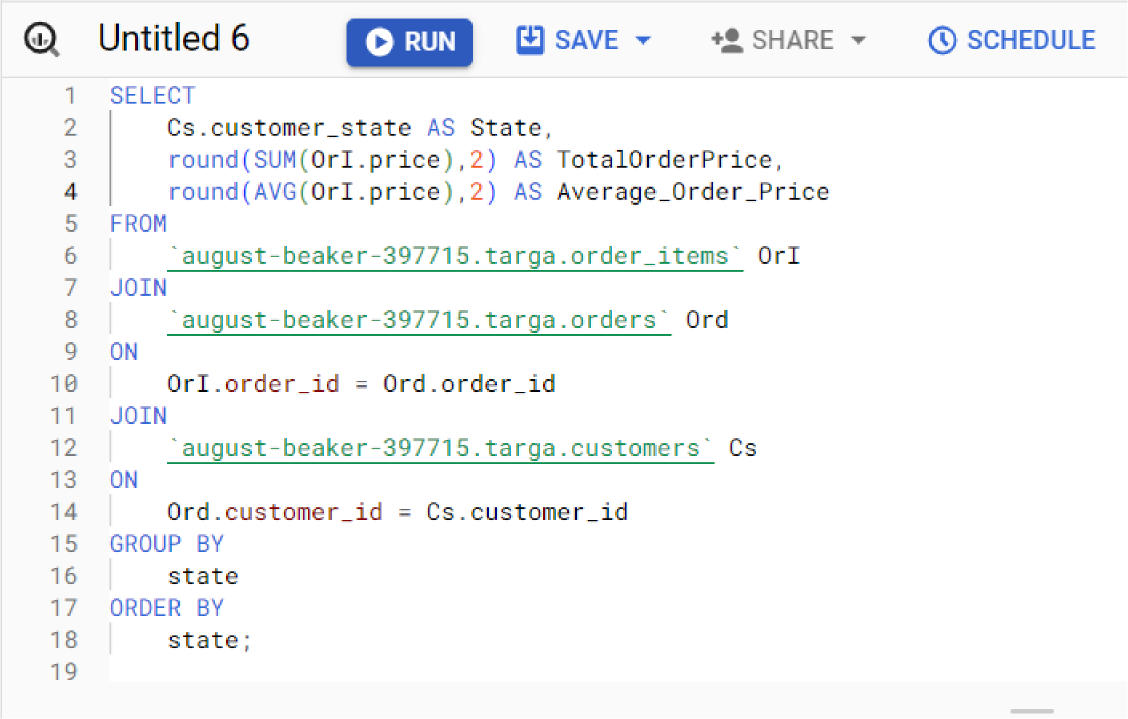
```
WITH base AS (
    SELECT
        EXTRACT(YEAR FROM TIMESTAMP(o.order_purchase_timestamp)) AS year,
        EXTRACT(MONTH FROM TIMESTAMP(o.order_purchase_timestamp)) AS month,
        SUM(p.payment_value) AS total_payment
    FROM
        `august-beaker-397715.targa.orders` o
    INNER JOIN
        `august-beaker-397715.targa.payments` p
    ON
        o.order_id = p.order_id
    WHERE
        EXTRACT(YEAR FROM TIMESTAMP(o.order_purchase_timestamp)) IN (2017, 2018)
        AND EXTRACT(MONTH FROM TIMESTAMP(o.order_purchase_timestamp)) BETWEEN 1 AND 8
    GROUP BY
        year, month
    ORDER BY
        year, month
),
YearlyComparison AS (
    SELECT
        year,
        month,
        total_payment,
        LAG(total_payment) OVER (ORDER BY year, month) AS prev_year_payment
    FROM
        base
    WHERE
        year IN (2017, 2018)
)

SELECT
    year,
    month,
    total_payment,
    CONCAT(ROUND(((total_payment - prev_year_payment) / prev_year_payment) *
100, 2), '%') AS percentage_increase
FROM
    YearlyComparison
WHERE
    year = 2018;
```



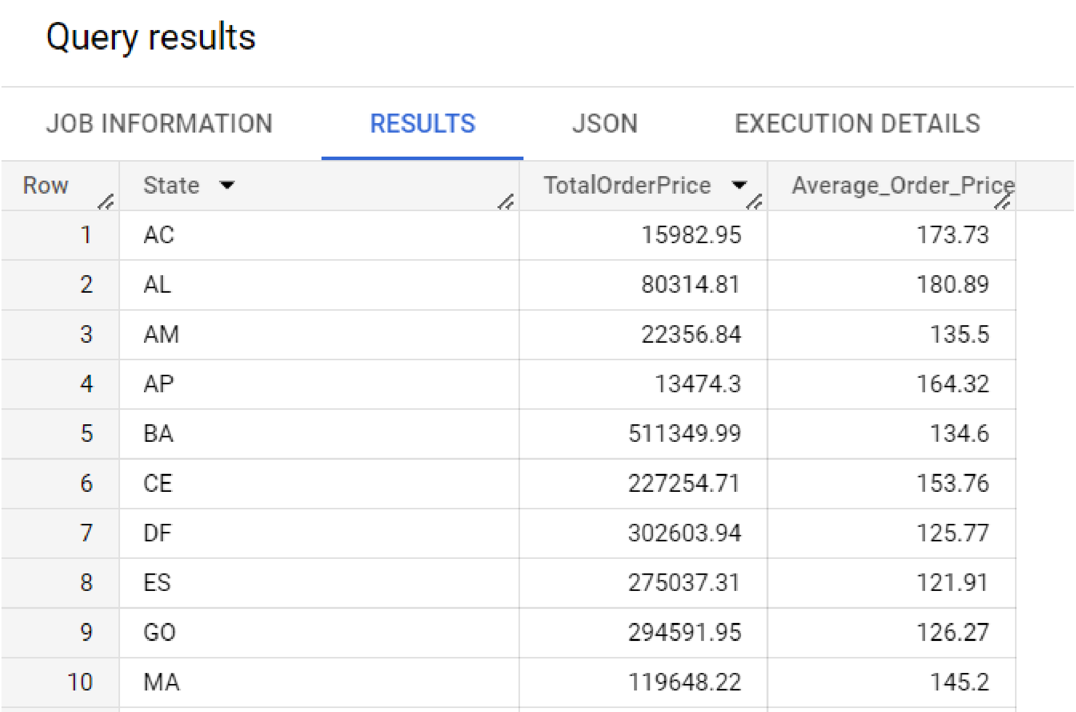
Row	year	month	total_payment	percentage_increase
1	2018	3	1159652.119999...	16.85%
2	2018	6	1023880.499999...	-11.27%
3	2018	7	1066540.750000...	4.17%
4	2018	8	1022425.320000...	-4.14%
5	2018	1	1115004.180000...	65.33%
6	2018	4	1160785.479999...	0.1%
7	2018	5	1153982.149999...	-0.59%
8	2018	2	992463.3400000...	-10.99%

Insight:- we have find percentage increase in year 2017 to 2018 we find that in month 1 there is 65 percentage increase and in some months there was decrease

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

```sql
SELECT
    Cs.customer_state AS State,
    round(SUM(OrI.price),2) AS TotalOrderPrice,
    round(AVG(OrI.price),2) AS Average_Order_Price
FROM
    `august-beaker-397715.targa.order_items` OrI
JOIN
    `august-beaker-397715.targa.orders` Ord
ON
    OrI.order_id = Ord.order_id
JOIN
    `august-beaker-397715.targa.customers` Cs
ON
    Ord.customer_id = Cs.customer_id
GROUP BY
    state
ORDER BY
    state;

```

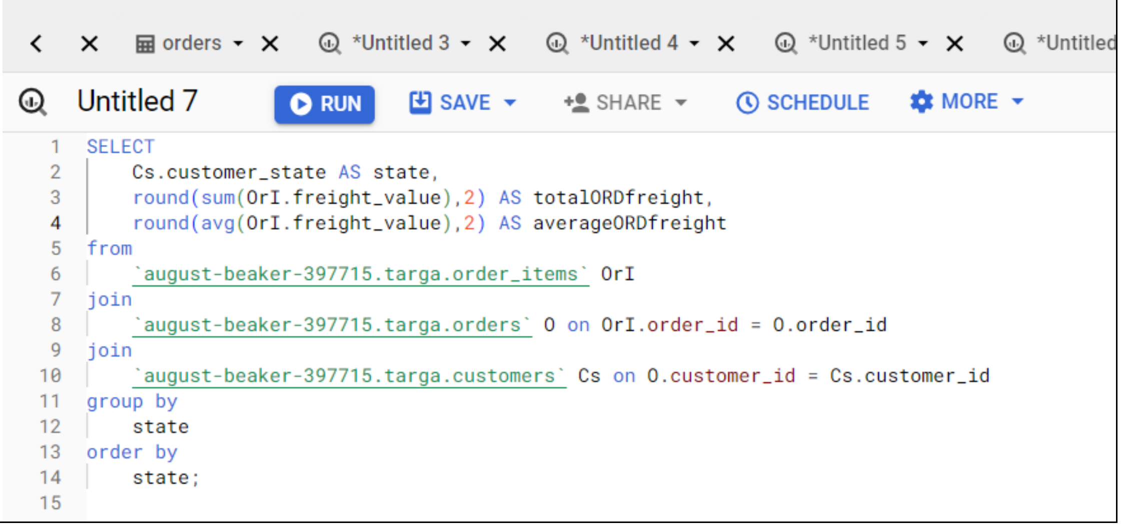


Row	State	TotalOrderPrice	Average_Order_Price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

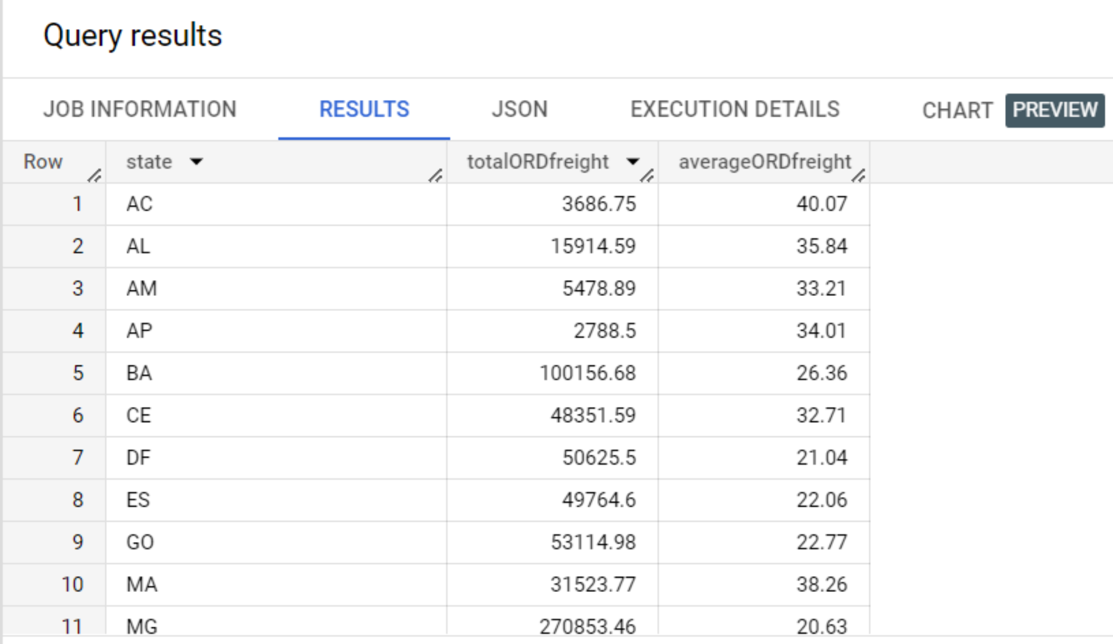
Insight :- here we found the average order price and the total order price for each state and we sorted by ascending order of basis of states

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

4.3



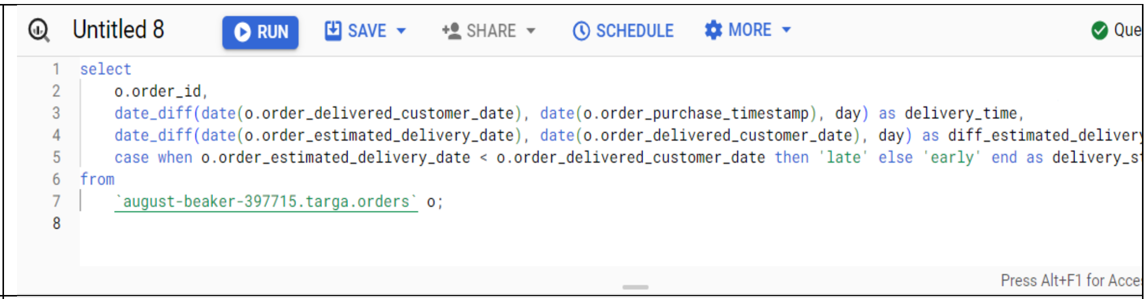
```
SELECT
    Cs.customer_state AS state,
    round(sum(OrI.freight_value),2) AS totalORDfreight,
    round(avg(OrI.freight_value),2) AS averageORDfreight
from
    `august-beaker-397715.targa.order_items` OrI
join
    `august-beaker-397715.targa.orders` O on OrI.order_id = O.order_id
join
    `august-beaker-397715.targa.customers` Cs on O.customer_id = Cs.customer_id
group by
    state
order by
    state;
```

<table>
<tr><td></td><td>


<table>
<tr><th>Row</th><th>state</th><th>totalORDfreight</th><th>averageORDfreight</th></tr>
<tr><td>1</td><td>AC</td><td>3686.75</td><td>40.07</td></tr>
<tr><td>2</td><td>AL</td><td>15914.59</td><td>35.84</td></tr>
<tr><td>3</td><td>AM</td><td>5478.89</td><td>33.21</td></tr>
<tr><td>4</td><td>AP</td><td>2788.5</td><td>34.01</td></tr>
<tr><td>5</td><td>BA</td><td>100156.68</td><td>26.36</td></tr>
<tr><td>6</td><td>CE</td><td>48351.59</td><td>32.71</td></tr>
<tr><td>7</td><td>DF</td><td>50625.5</td><td>21.04</td></tr>
<tr><td>8</td><td>ES</td><td>49764.6</td><td>22.06</td></tr>
<tr><td>9</td><td>GO</td><td>53114.98</td><td>22.77</td></tr>
<tr><td>10</td><td>MA</td><td>31523.77</td><td>38.26</td></tr>
<tr><td>11</td><td>MG</td><td>270853.46</td><td>20.63</td></tr>
</table>
</td></tr>
<tr><td></td><td>Insight :- here we found the average order frieght and the total order frieght for each state and we sorted by ascending order of basis of states</td></tr>
</table>

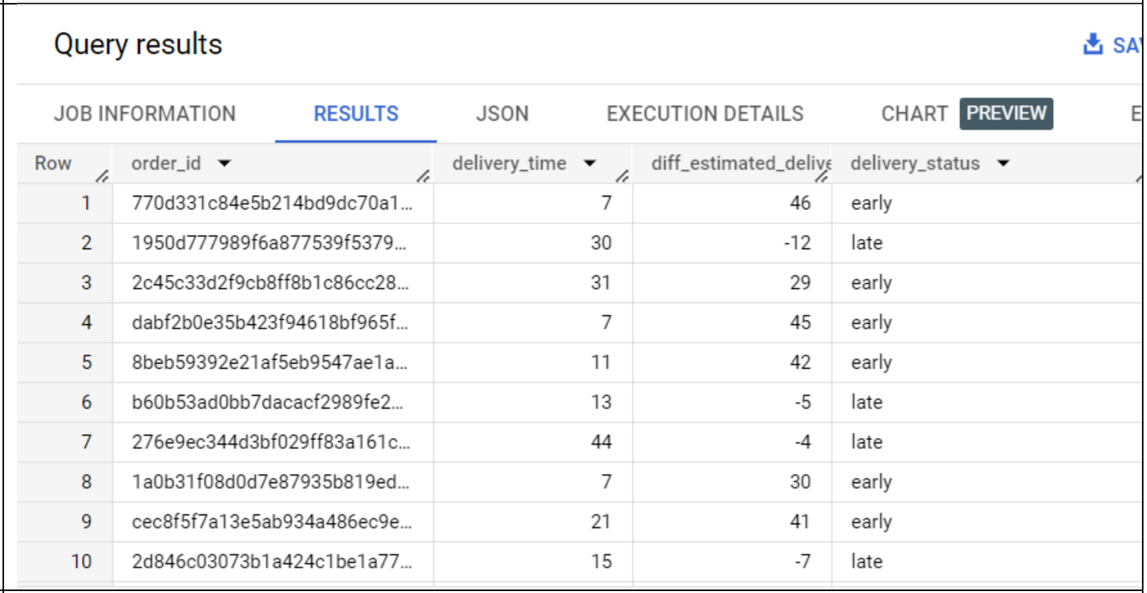
5 Analysis based on sales, freight and delivery time.

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
    o.order_id,
    date_diff(date(o.order_delivered_customer_date), date(o.order_purchase_timestamp), day) as delivery_time,
    date_diff(date(o.order_estimated_delivery_date), date(o.order_delivered_customer_date), day) as diff_estimated_deliver
    case when o.order_estimated_delivery_date < o.order_delivered_customer_date then 'late' else 'early' end as delivery_s
from
    `august-beaker-397715.targa.orders` o;

```



Row	order_id	delivery_time	diff_estimated_delive	delivery_status
1	770d331c84e5b214bd9dc70a1...	7	46	early
2	1950d777989f6a877539f5379...	30	-12	late
3	2c45c33d2f9cb8ff8b1c86cc28...	31	29	early
4	dabf2b0e35b423f94618bf965f...	7	45	early
5	8beb59392e21af5eb9547ae1a...	11	42	early
6	b60b53ad0bb7dacacf2989fe2...	13	-5	late
7	276e9ec344d3bf029ff83a161c...	44	-4	late
8	1a0b31f08d0d7e87935b819ed...	7	30	early
9	cec8f5f7a13e5ab934a486ec9e...	21	41	early
10	2d846c03073b1a424c1be1a77...	15	-7	late

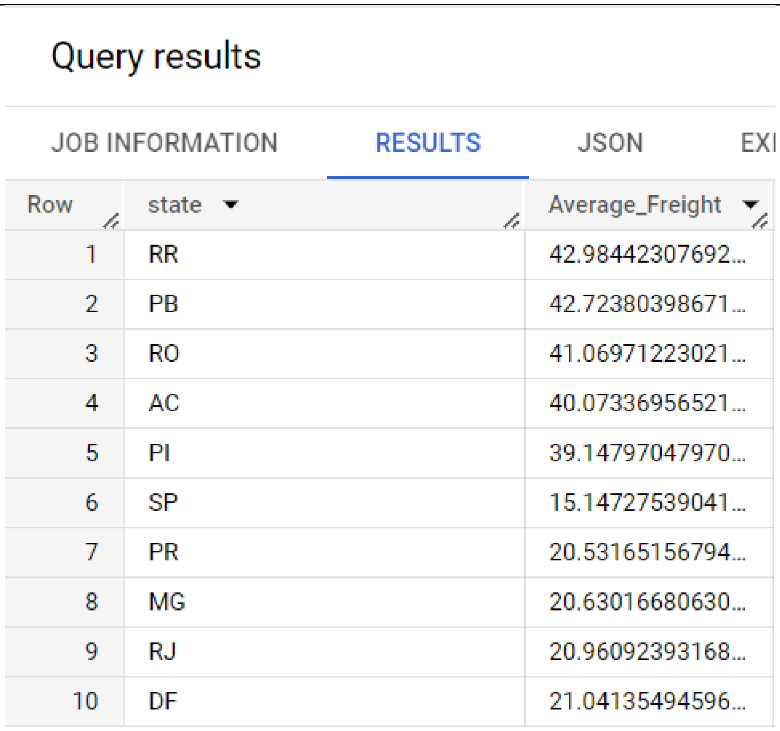
Here we calculated the delivery time taken and the estimated delivery difference using the date diff function and additionaly we also find the delivery status as delivered early or late

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

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

```
with stateaveragefreight as (
    select c.customer_state as state,avg(oi.freight_value) as Average_Freight
    from `august-beaker-397715.targa.order_items` oi
    join `august-beaker-397715.targa.orders` o on oi.order_id = o.order_id
    join `august-beaker-397715.targa.customers` c on o.customer_id =
c.customer_id
    group by state
)
(
select state,Average_Freight
from stateaveragefreight
order by Average_Freight desc
limit 5
)
```

```
union all
(
select state,Average_Freight from stateaveragefreight
order by Average_Freight asc
limit 5
 );
```



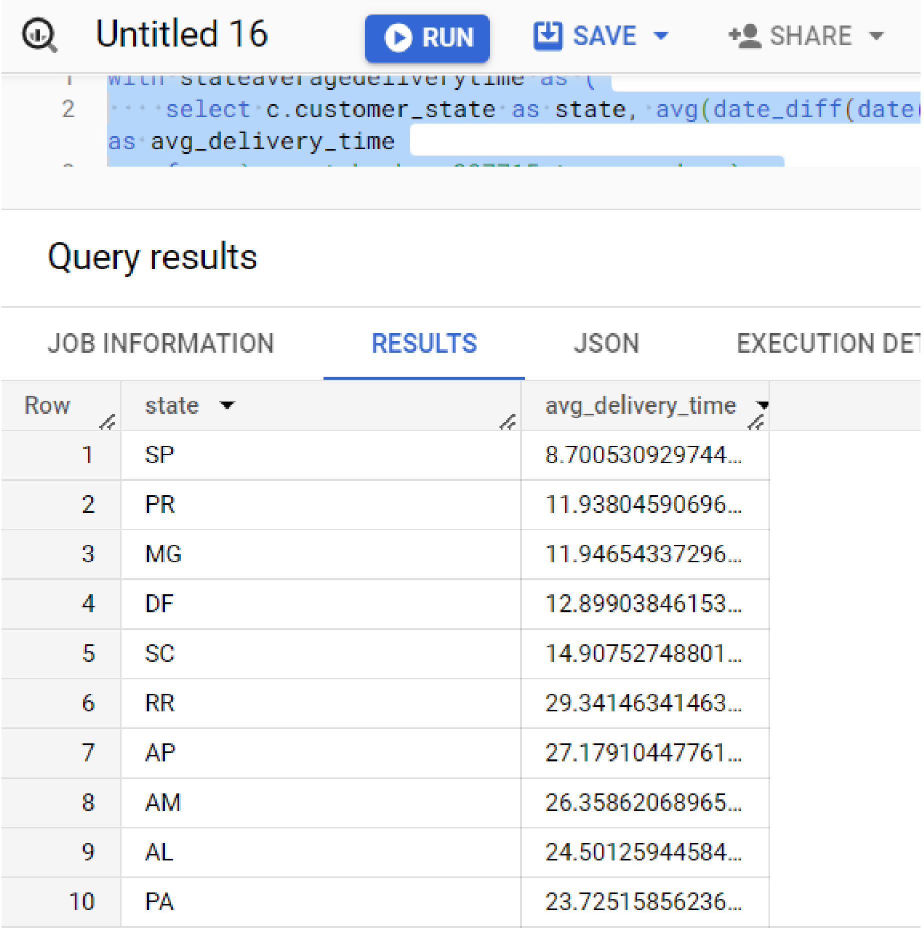
Row	state	Average_Freight
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...
6	SP	15.14727539041...
7	PR	20.53165156794...
8	MG	20.63016680630...
9	RJ	20.96092393168...
10	DF	21.04135494596...

Top 5 states with highest and lowest average freight frist 5 are highest states and last 5 are the lowest average freight state

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

5.3

```
with stateaveragedeliverytime as (
    select c.customer_state as state,
avg(date_diff(date(o.order_delivered_customer_date),
date(o.order_purchase_timestamp), day)) as avg_delivery_time
    from `august-beaker-397715.targa.orders` o
    join `august-beaker-397715.targa.customers` c
    on o.customer_id = c.customer_id
    where o.order_delivered_customer_date is not null
    group by state
)
(
    select state, avg_delivery_time
    from stateaveragedeliverytime
    order by avg_delivery_time desc
    limit 5
)
union all
(
    select state, avg_delivery_time
    from stateaveragedeliverytime
    order by avg_delivery_time asc
    limit 5
);
```



```
with stateaveragedeliverytime as (
    select c.customer_state as state, avg(date_diff(date
as avg_delivery_time
```

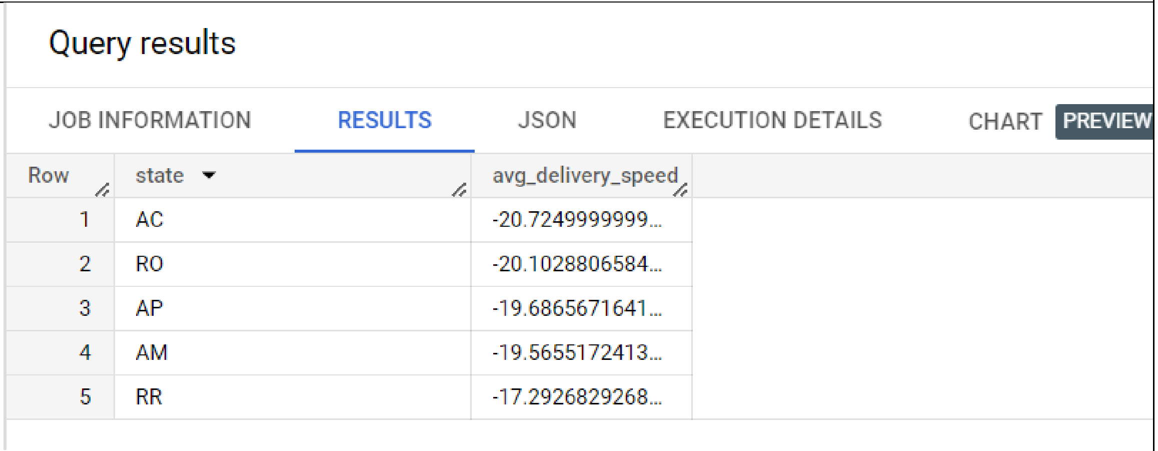
Query results

Row	state	avg_delivery_time
1	SP	8.700530929744...
2	PR	11.93804590696...
3	MG	11.94654337296...
4	DF	12.89903846153...
5	SC	14.90752748801...
6	RR	29.34146341463...
7	AP	27.17910447761...
8	AM	26.35862068965...
9	AL	24.50125944584...
10	PA	23.72515856236...

insight	Top 5 states with highest and lowest average delivery time taken frist 5 are highest states with fastest delivery and last 5 are the lowest on time delivery state

5.4 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.

```
with statedeliveryspeed as (
    select c.customer_state as
state,avg(date_diff(date(o.order_delivered_customer_date),
date(o.order_estimated_delivery_date), day)) as avg_delivery_speed
    from `august-beaker-397715.targa.orders` o
    join `august-beaker-397715.targa.customers` c on o.customer_id = c.customer_id
    group by state)
select state,avg_delivery_speed from statedeliveryspeed
where avg_delivery_speed < 0
order by avg_delivery_speed asc
limit 5
```



Row	state	avg_delivery_speed
1	AC	-20.7249999999...
2	RO	-20.1028806584...
3	AP	-19.6865671641...
4	AM	-19.5655172413...
5	RR	-17.2926829268...

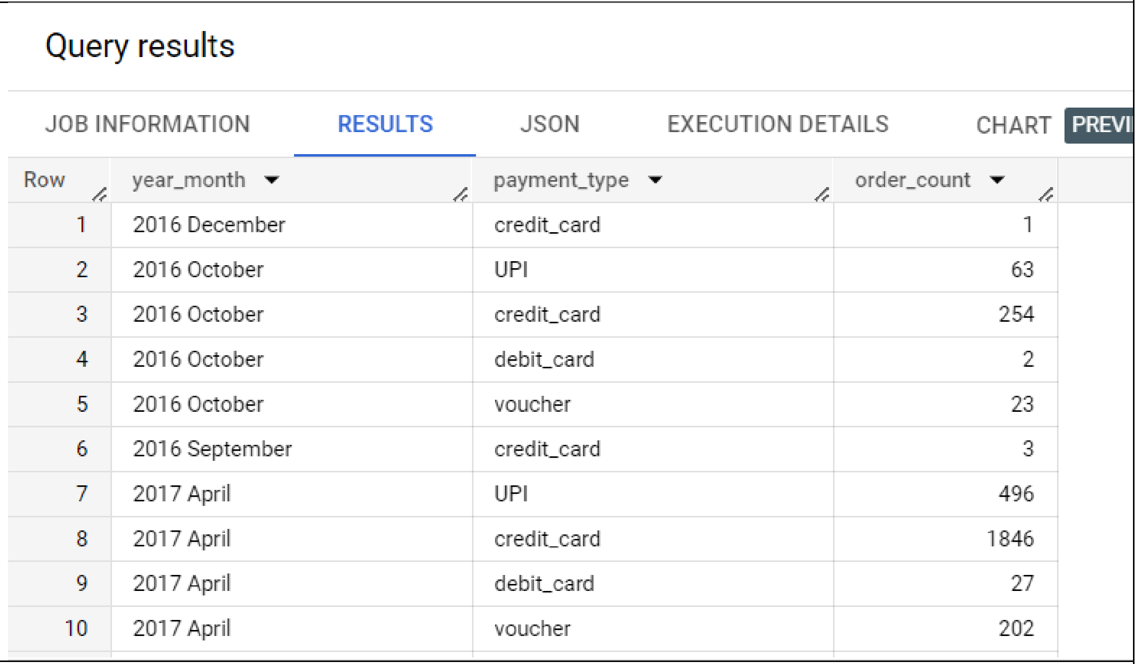
Insights:- We found this states where the order delivery is fast this shows that AC state may have good logistics and we found this using date diff function between orderdelivered date and expected delivery date

6 Analysis based on the payments:

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

```
select concat( extract(year from timestamp(o.order_purchase_timestamp)),' ',
format_timestamp('%B', timestamp(o.order_purchase_timestamp))) as
year_month,p.payment_type,
count(o.order_id) as order_count
from `august-beaker-397715.targa.orders` o
```

```
join `august-beaker-397715.targa.payments` p on o.order_id = p.order_id
group by year_month,payment_type
order by year_month, payment_type
```



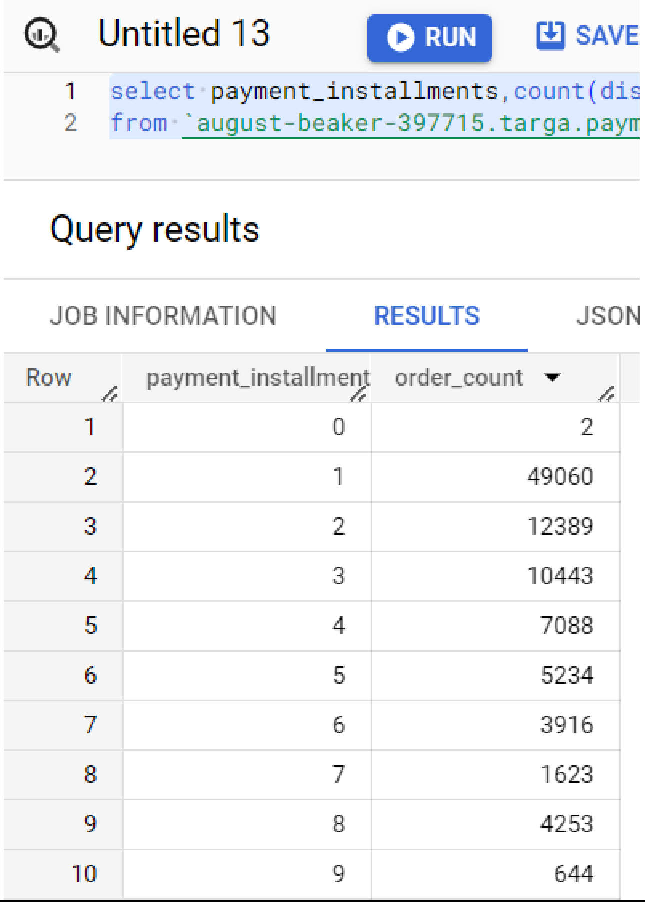
Query results

Row	year_month	payment_type	order_count
1	2016 December	credit_card	1
2	2016 October	UPI	63
3	2016 October	credit_card	254
4	2016 October	debit_card	2
5	2016 October	voucher	23
6	2016 September	credit_card	3
7	2017 April	UPI	496
8	2017 April	credit_card	1846
9	2017 April	debit_card	27
10	2017 April	voucher	202

Here we find that how many orders where there for particular payment type as you can see most Brazilian used credit cards and we extracted this data on monthly basis.

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

```
select payment_installments,count(distinct order_id) as order_count
from `august-beaker-397715.targa.payments`
group by payment_installments
order by payment_installments;
```



```
select payment_installments,count(dis
from `august-beaker-397715.targa.paym
```

Row	payment_installment	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

Insights :- Here we found the payment installments that is how much installments and how many orders were placed, such as installment 1 is there so 49060 orders were there in which the installment was 1.