Business Case: Target SQL

Problem Statement: Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analysing the given dataset to extract valuable insights and provide actionable recommendations.

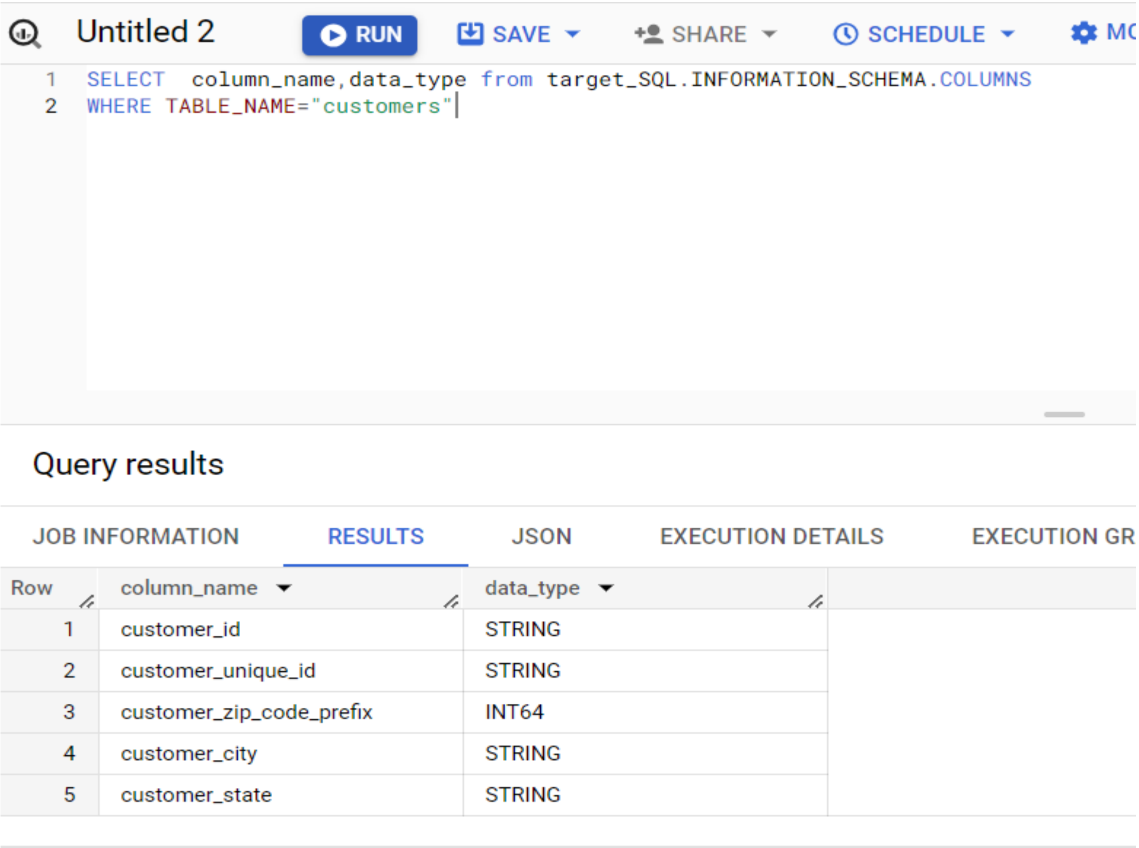
What does 'good' look like?

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.

Query:

```
SELECT  column_name,data_type from target_SQL.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME="customers"
```



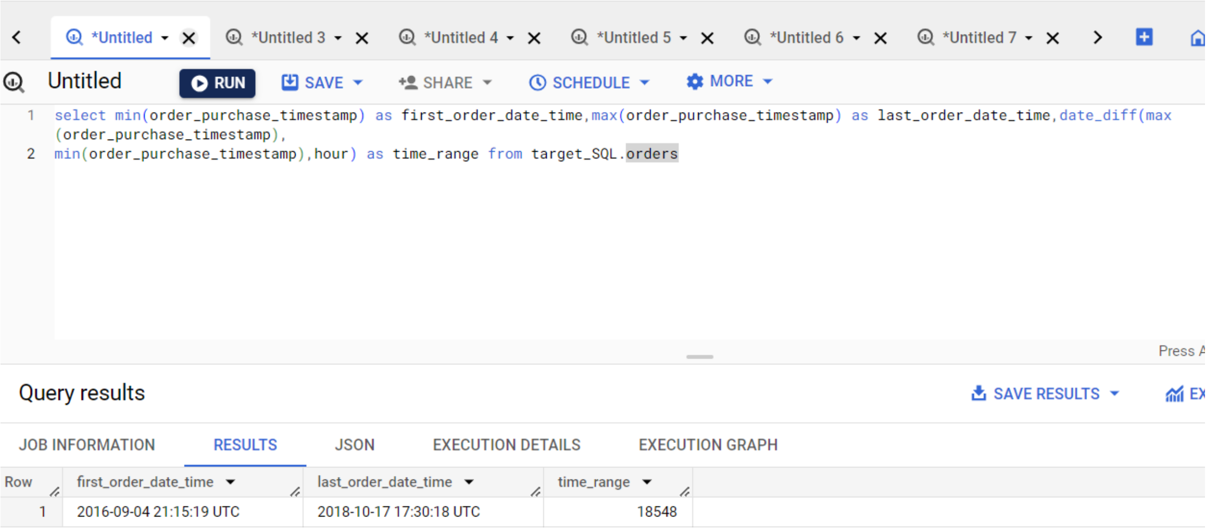
```
SELECT  column_name,data_type from target_SQL.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME="customers"
```

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

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

Query:

select min(order_purchase_timestamp) as first_order_date_time, max(order_ purchase_ timestamp) as last_order_date_time,date_diff(max(order_purchase_timestamp), min(order_purchase_timestamp),hour) as time_range from target_SQL.orders



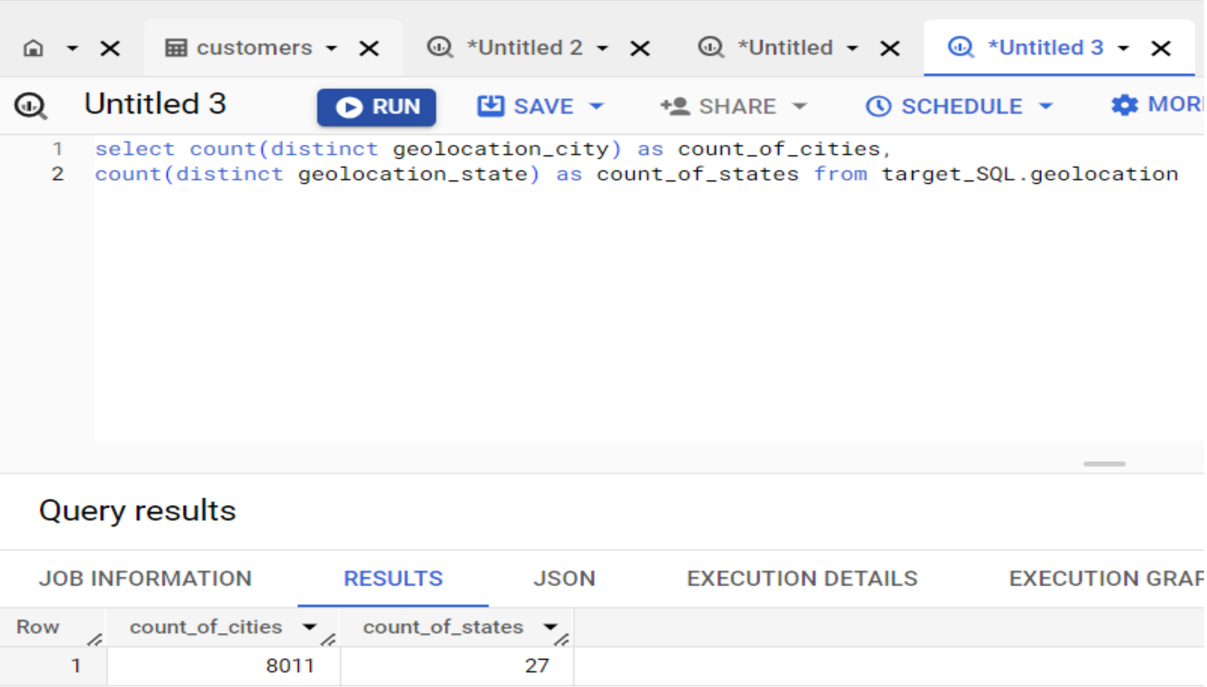
```
select min(order_purchase_timestamp) as first_order_date_time,max(order_purchase_timestamp) as last_order_date_time,date_diff(max
(order_purchase_timestamp),
min(order_purchase_timestamp),hour) as time_range from target_SQL.orders
```

Row	first_order_date_time	last_order_date_time	time_range
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	18548

3.Count the number of Cities and States in our dataset.

Query:

select count(distinct geolocation_city) as count_of_cities, count(distinct geolocation_state) as count_of_states from target_SQL.geolocation



```
select count(distinct geolocation_city) as count_of_cities,
count(distinct geolocation_state) as count_of_states from target_SQL.geolocation
```

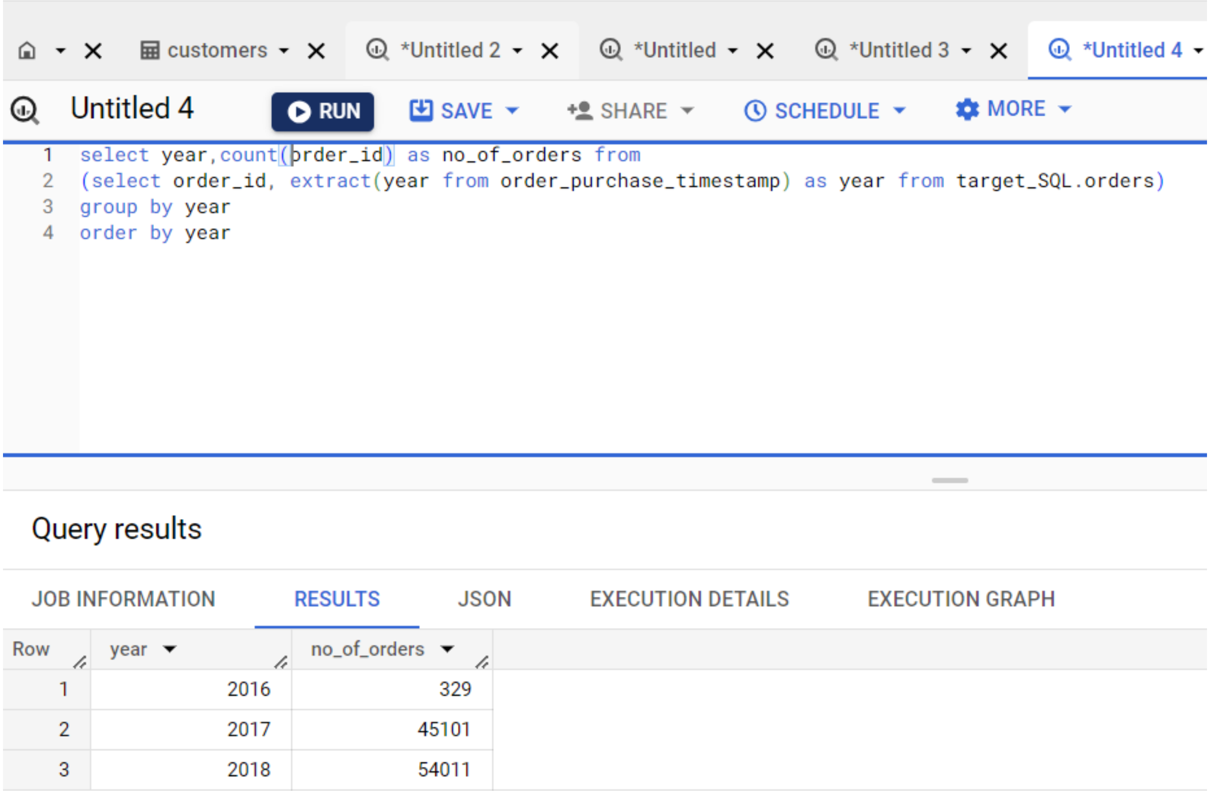
Row	count_of_cities	count_of_states
1	8011	27

2.In-depth Exploration:

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

Query:

select year,count(order_id) as no_of_orders from (select order_id, extract(year from order_purchase_timestamp) as year from target_SQL.orders) group by year order by year



```
select year,count(order_id) as no_of_orders from
(select order_id, extract(year from order_purchase_timestamp) as year from target_SQL.orders)
group by year
order by year
```

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

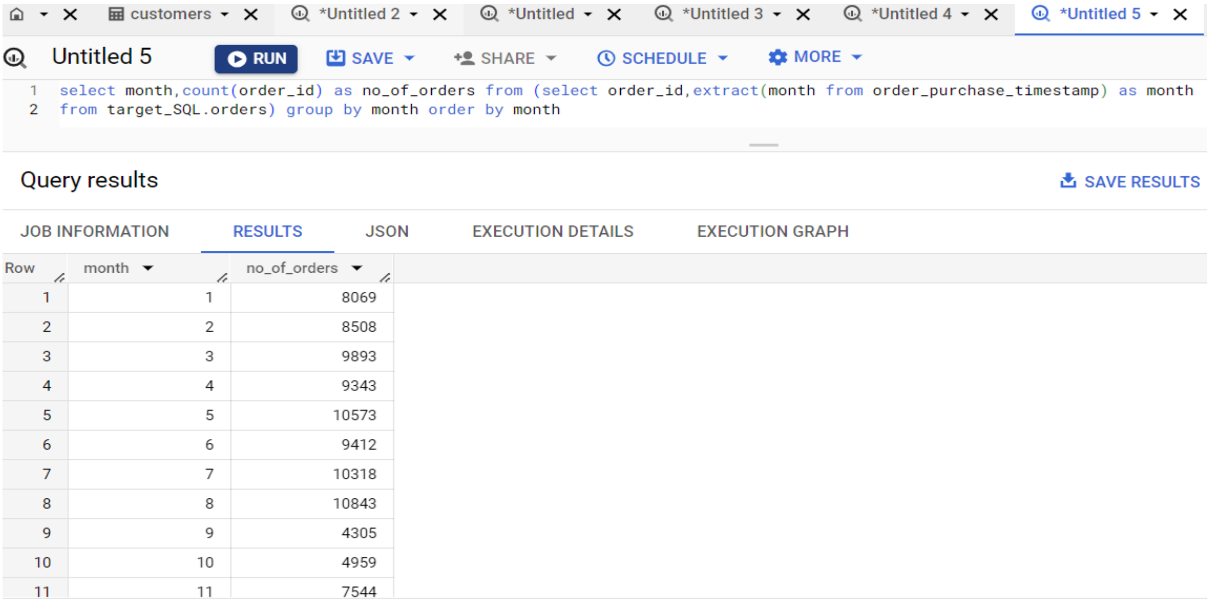
Insights:

- From the results above, we can clearly see an increase in the number of orders over the years.
- Number of orders has increased from 329 in 2016 to 45101 in 2017 to 54011 in 2018

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

Query:

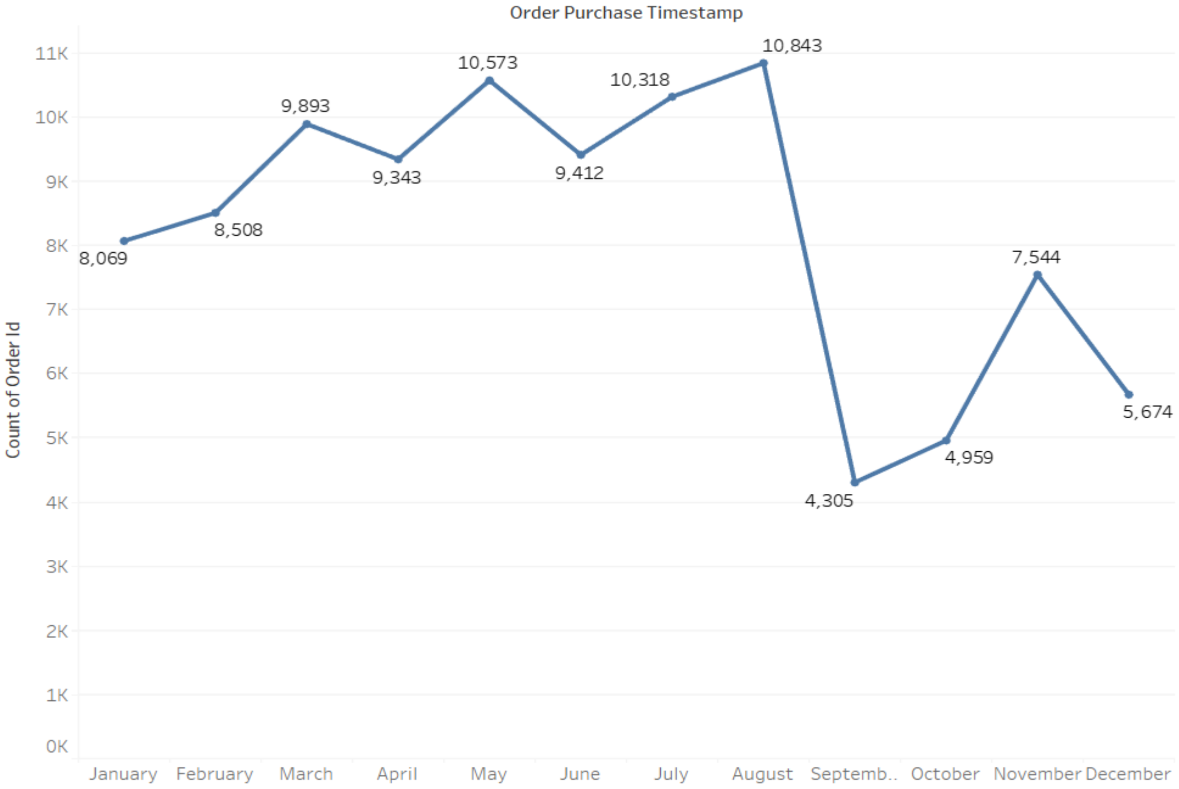
```
select month,count(order_id) as no_of_orders from (select order_id,extract(month from

order_purchase_timestamp) as month from target_SQL.orders) group by month order by

 month
```



```
select month,count(order_id) as no_of_orders from (select order_id,extract(month from order_purchase_timestamp) as month
from target_SQL.orders) group by month order by month
```

Row	month	no_of_orders
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

Sheet 1



Insights:

- From the above graph, we can see that in the month of May, July and August , number of orders has increased
- In the month of September, October and December , number of orders has decreased

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

Query:

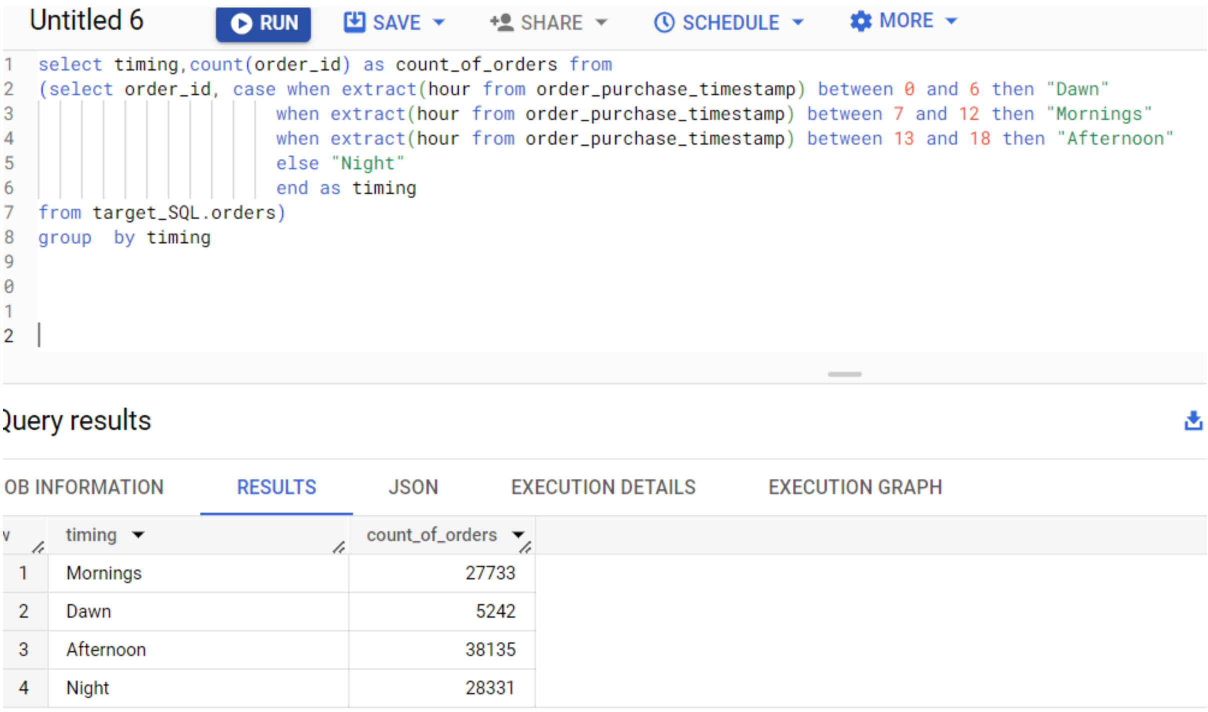
select timing,count(order_id) as count_of_orders from (select order_id,

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 timing from target_SQL.orders) group by timing



```
select timing,count(order_id) as count_of_orders from
(select order_id, 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 timing
from target_SQL.orders)
group  by timing
```

Query results

v	timing	count_of_orders
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

Insights:

- From the above results, we can see that most of the orders are placed during afternoon

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

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

Query:

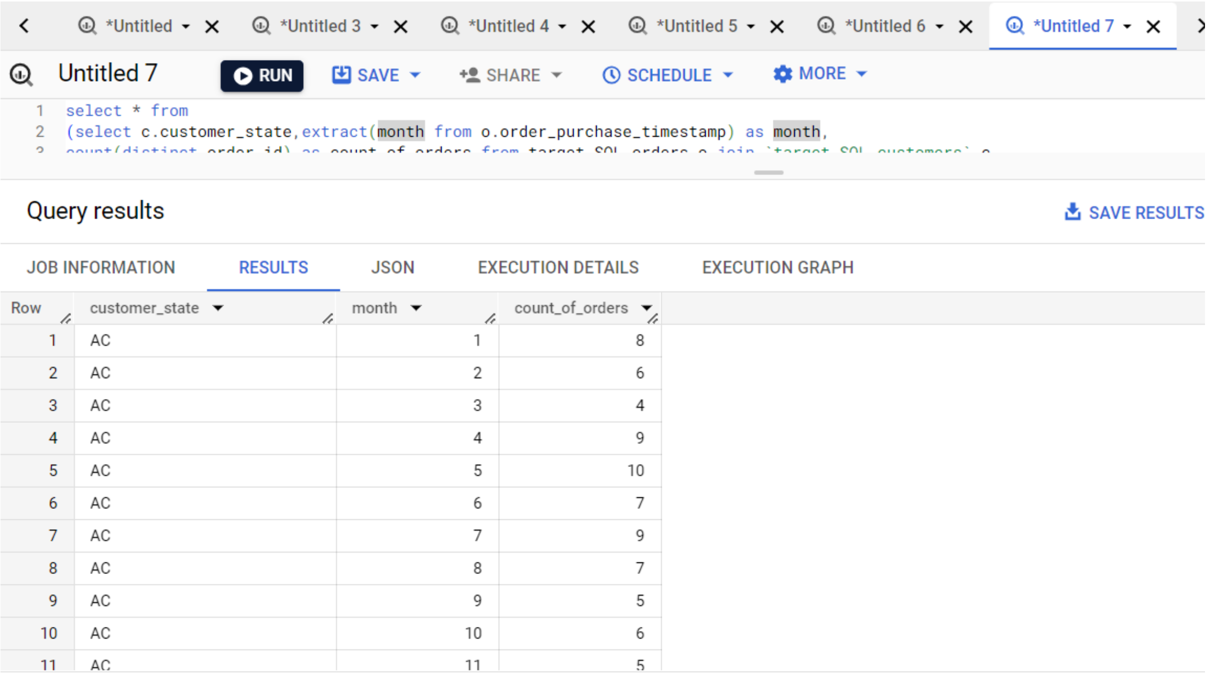
select * from (select c.customer_state,extract(month from o.order_purchase_timestamp)

as month,count(distinct order_id) as count_of_orders from target_SQL.orders o join

`target_SQL.customers` c on o.customer_id=c.customer_id group by

c.customer_state,extract(month from o.order_purchase_timestamp)) order by

customer_state, month



```
select * from
(select c.customer_state,extract(month from o.order_purchase_timestamp) as month,
count(distinct order_id) as count_of_orders from target_SQL.orders o join `target_SQL.customers` c
```

Row	customer_state	month	count_of_orders
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

Insights:

- From the above results, we can see that there is no significant change in month on month number of orders for each year and each state.
- Except for MG , where slight increase in number of orders each month has been observed in 2017,whereas decrease in 2018.

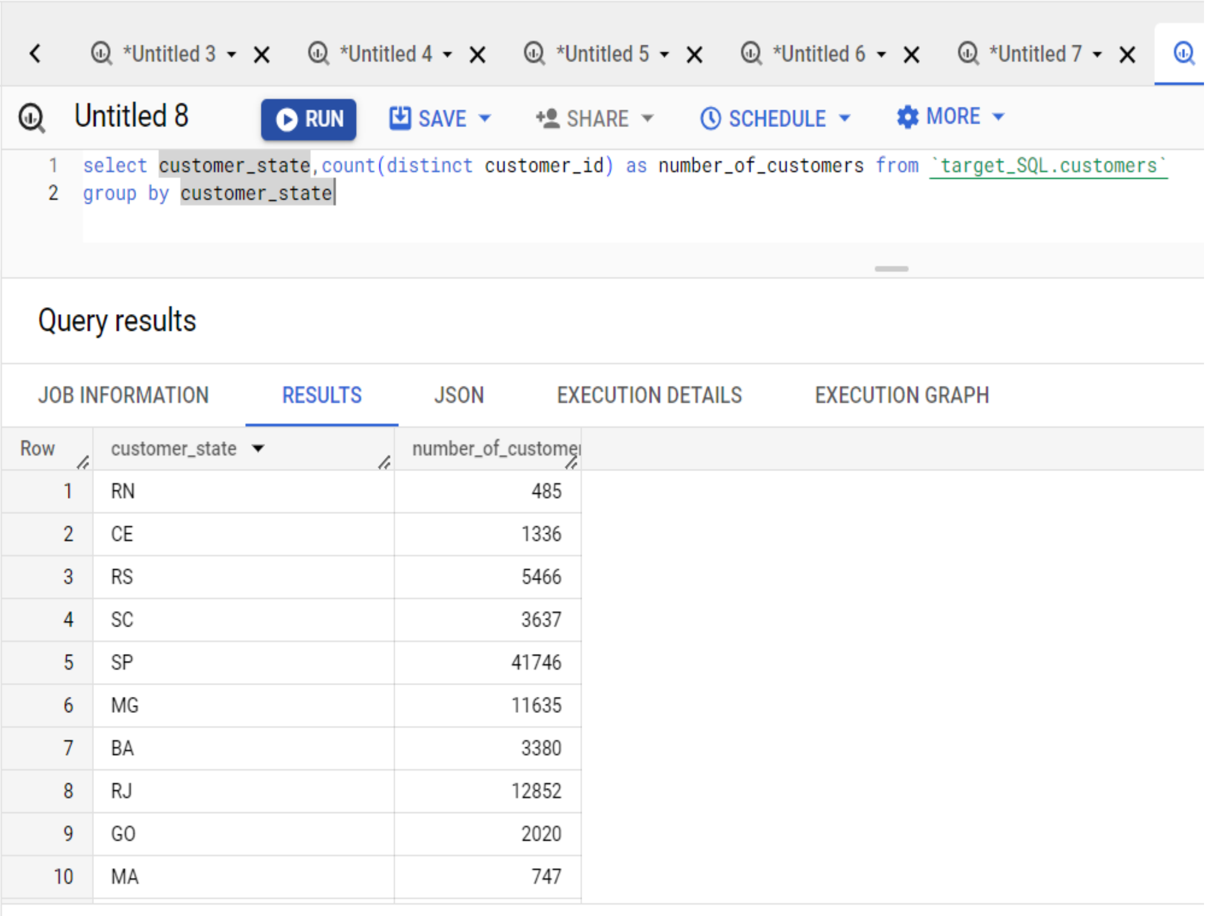
- For RJ as well there was slight increase in number of orders each month in 2017, whereas decrease in 2018.
- In SP , there was a significant increase in number of orders placed each month in 2017,and significant decrease after month of September 2018.

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

Query:

select customer_state,count(distinct customer_id) as number_of_customers from

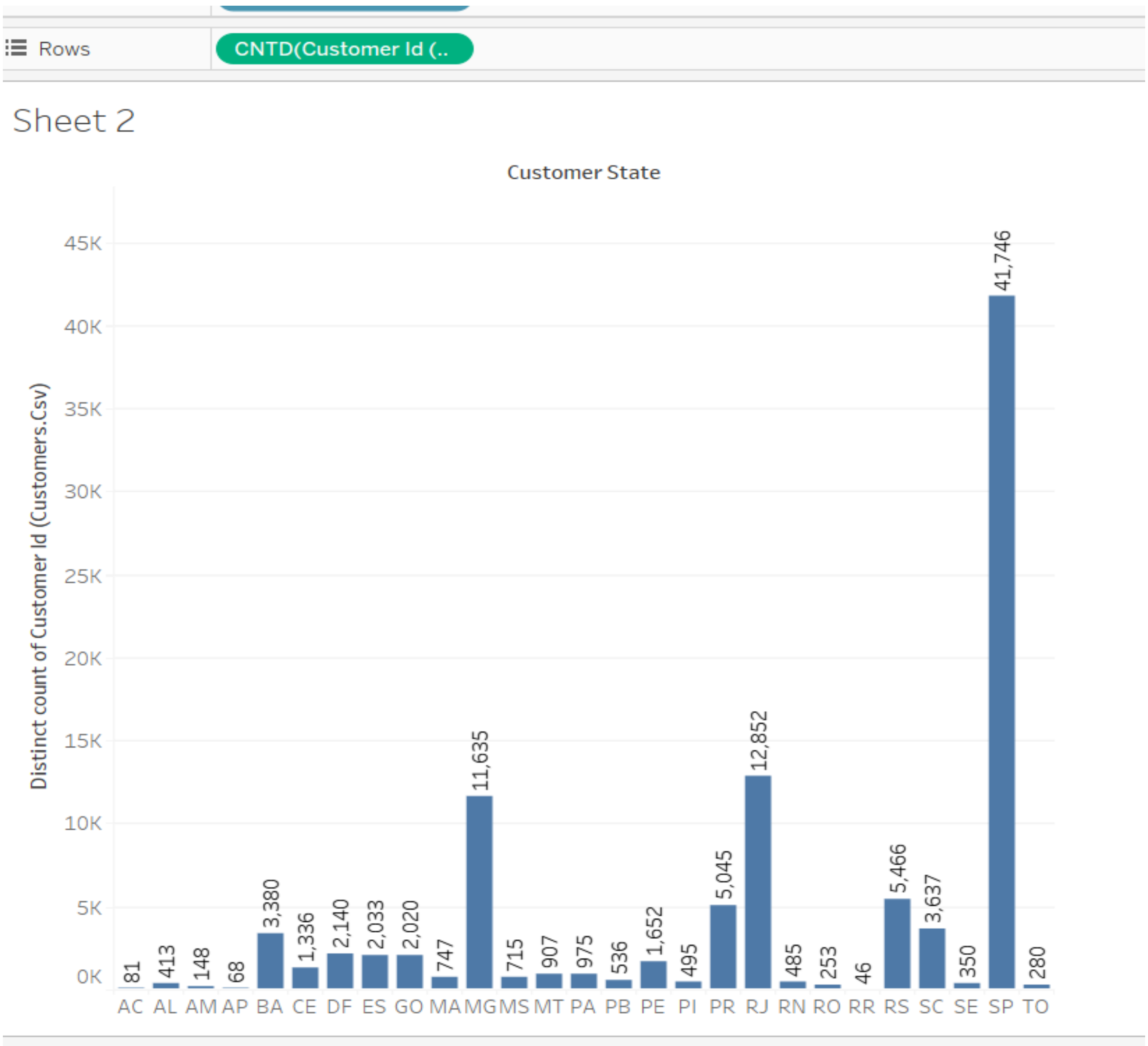
`target_SQL.customers` group by customer_state



```
select customer_state,count(distinct customer_id) as number_of_customers from `target_SQL.customers`
group by customer_state
```

Query results

Row	customer_state	number_of_custome
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747



Rows CNTD(Customer Id (..

Sheet 2

Insights:

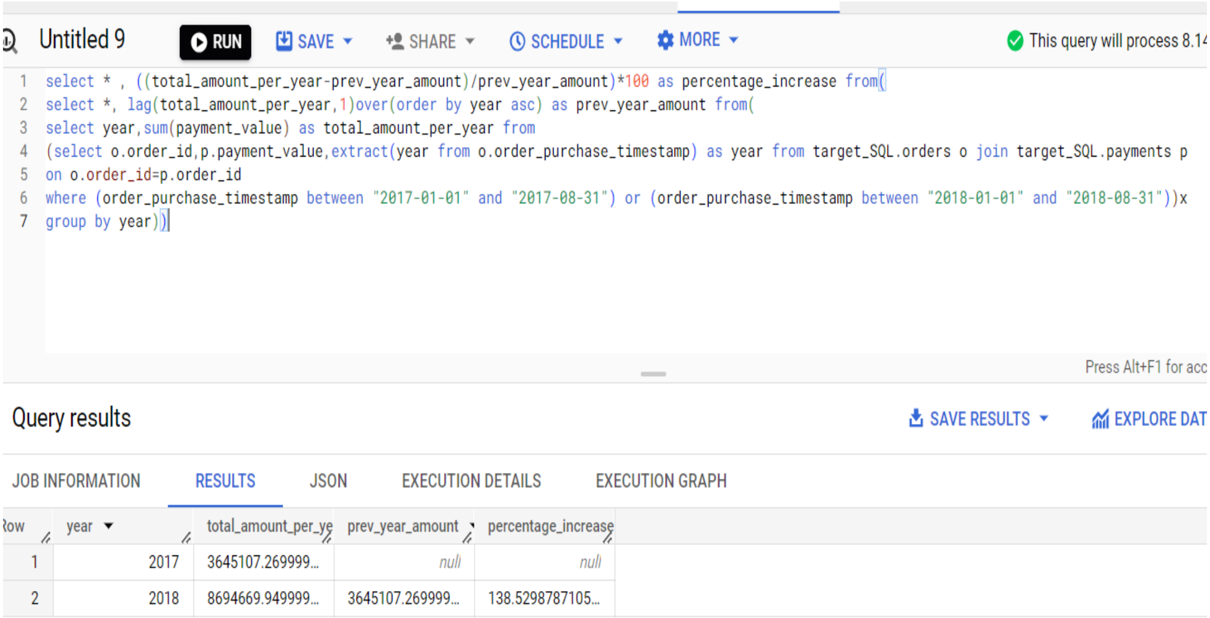
- From the above results , we can see that maximum number of customers are from SP.
- Lowest number of customers are from RR.

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).
You can use the "payment_value" column in the payments table to get the cost of orders.

Query:

```
select * , ((total_amount_per_year-prev_year_amount)/prev_year_amount)*100 as
percentage_increase from(select *, lag(total_amount_per_year,1)over(order by year asc) as
prev_year_amount from(select year,sum(payment_value) as total_amount_per_year from
(select o.order_id,p.payment_value,extract(year from o.order_purchase_timestamp) as
year from target_SQL.orders o join target_SQL.payments p on o.order_id=p.order_id
where (order_purchase_timestamp between "2017-01-01" and "2017-08-31") or
(order_purchase_timestamp between "2018-01-01" and "2018-08-31"))x group by year))
```



```
select * , ((total_amount_per_year-prev_year_amount)/prev_year_amount)*100 as percentage_increase from(
select *, lag(total_amount_per_year,1)over(order by year asc) as prev_year_amount from(
select year,sum(payment_value) as total_amount_per_year from
(select o.order_id,p.payment_value,extract(year from o.order_purchase_timestamp) as year from target_SQL.orders o join target_SQL.payments p
on o.order_id=p.order_id
where (order_purchase_timestamp between "2017-01-01" and "2017-08-31") or (order_purchase_timestamp between "2018-01-01" and "2018-08-31"))x
group by year))
```

Row	year	total_amount_per_ye	prev_year_amount	percentage_increase
1	2017	3645107.269999...	null	null
2	2018	8694669.949999...	3645107.269999...	138.5298787105...

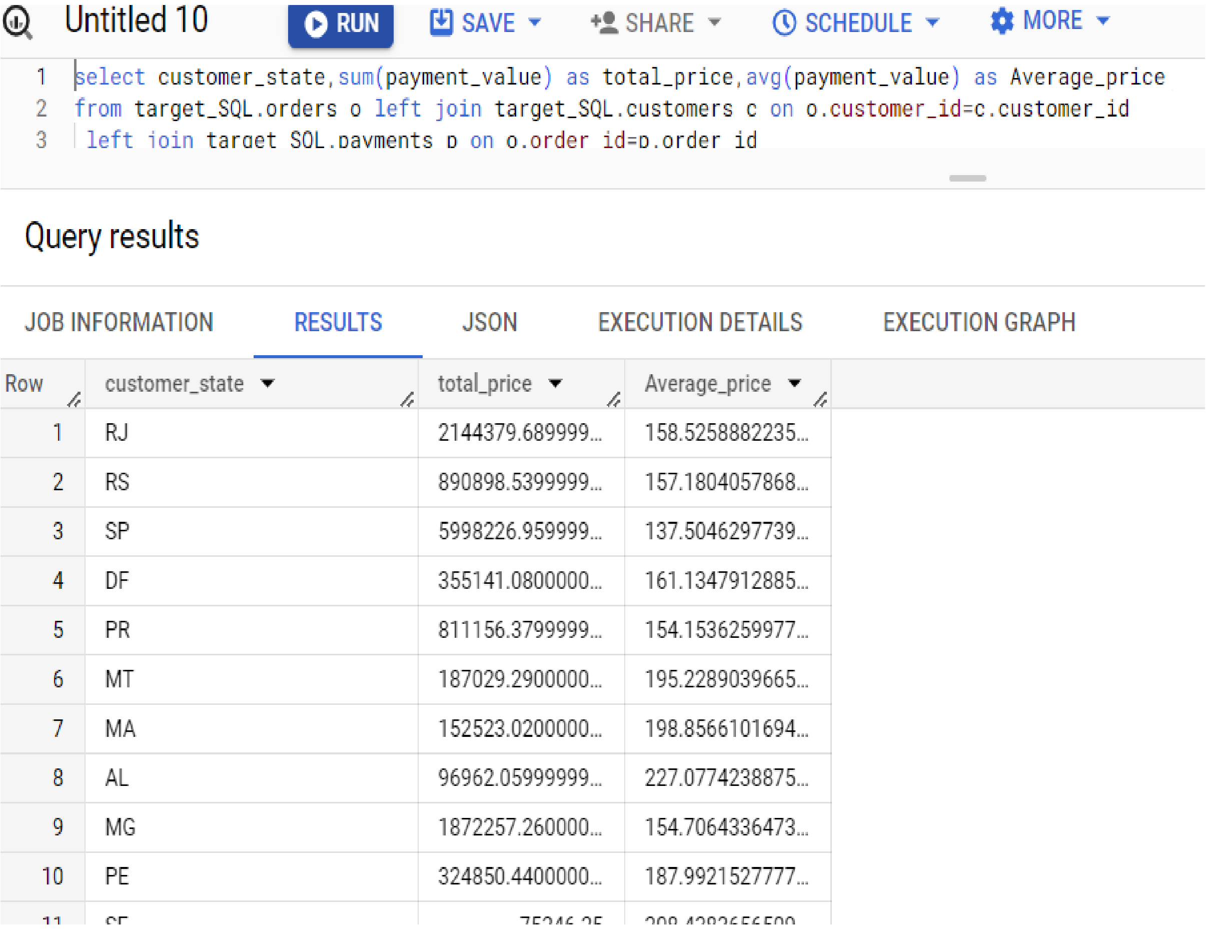
Insights:

- From the above results, we can see that the percentage increase in the cost of orders from 2017 to 2018 (including months between Jan to Aug only) is 138.52%

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

Query:

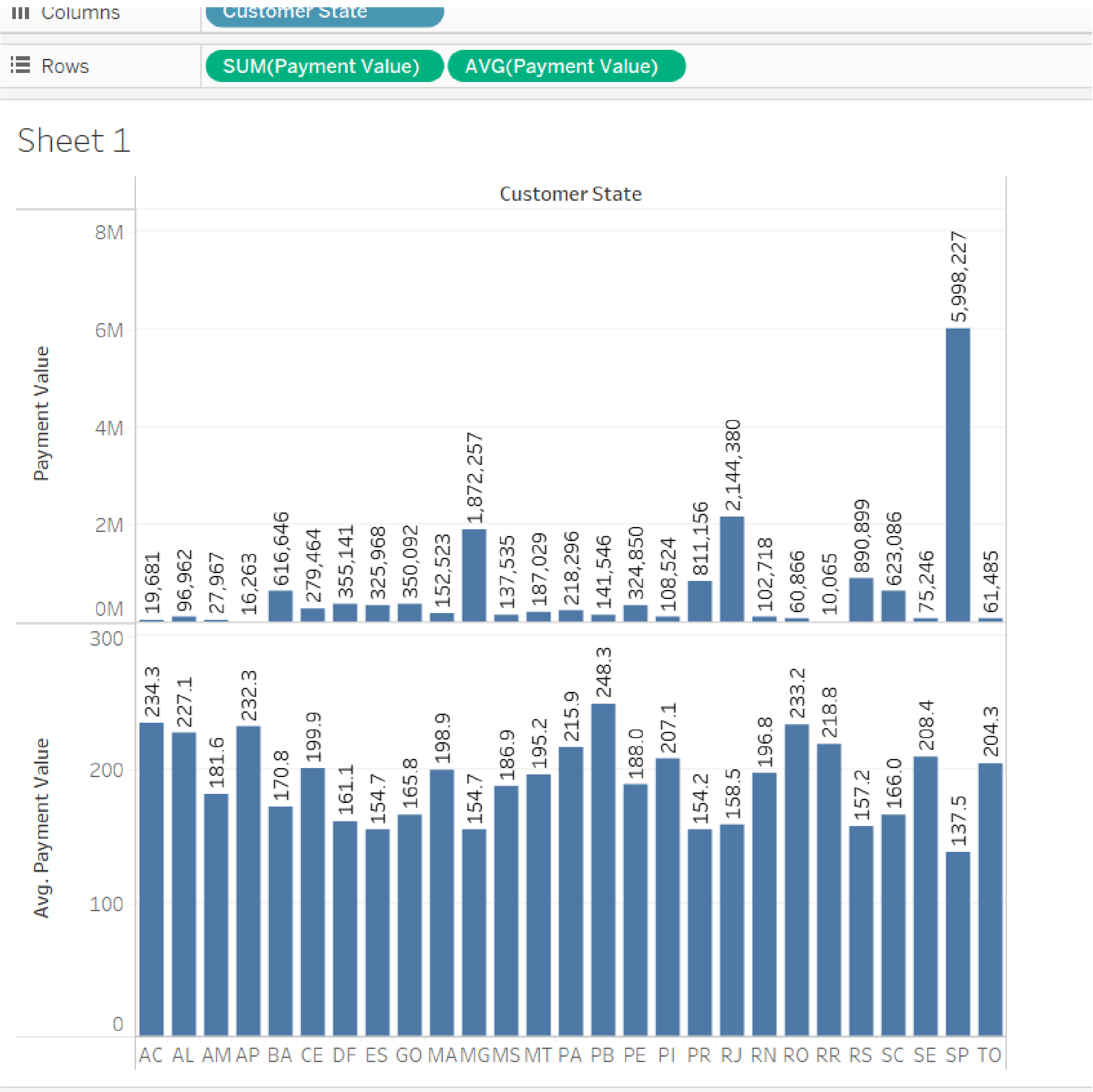
select customer_state,sum(payment_value) as total_price,avg(payment_value) as Average_price from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id left join target_SQL.payments p on o.order_id=p.order_id group by customer_state



```
select customer_state,sum(payment_value) as total_price,avg(payment_value) as Average_price
from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id
 left join target_SQL.payments p on o.order_id=p.order_id
```

Query results

Row	customer_state	total_price	Average_price
1	RJ	2144379.689999...	158.5258882235...
2	RS	890898.5399999...	157.1804057868...
3	SP	5998226.959999...	137.5046297739...
4	DF	355141.0800000...	161.1347912885...
5	PR	811156.3799999...	154.1536259977...
6	MT	187029.2900000...	195.2289039665...
7	MA	152523.0200000...	198.8566101694...
8	AL	96962.05999999...	227.0774238875...
9	MG	1872257.260000...	154.7064336473...
10	PE	324850.4400000...	187.9921527777...
11	SE	75246.25	208.4282656500



Columns: Customer State

Rows: SUM(Payment Value) AVG(Payment Value)

Sheet 1

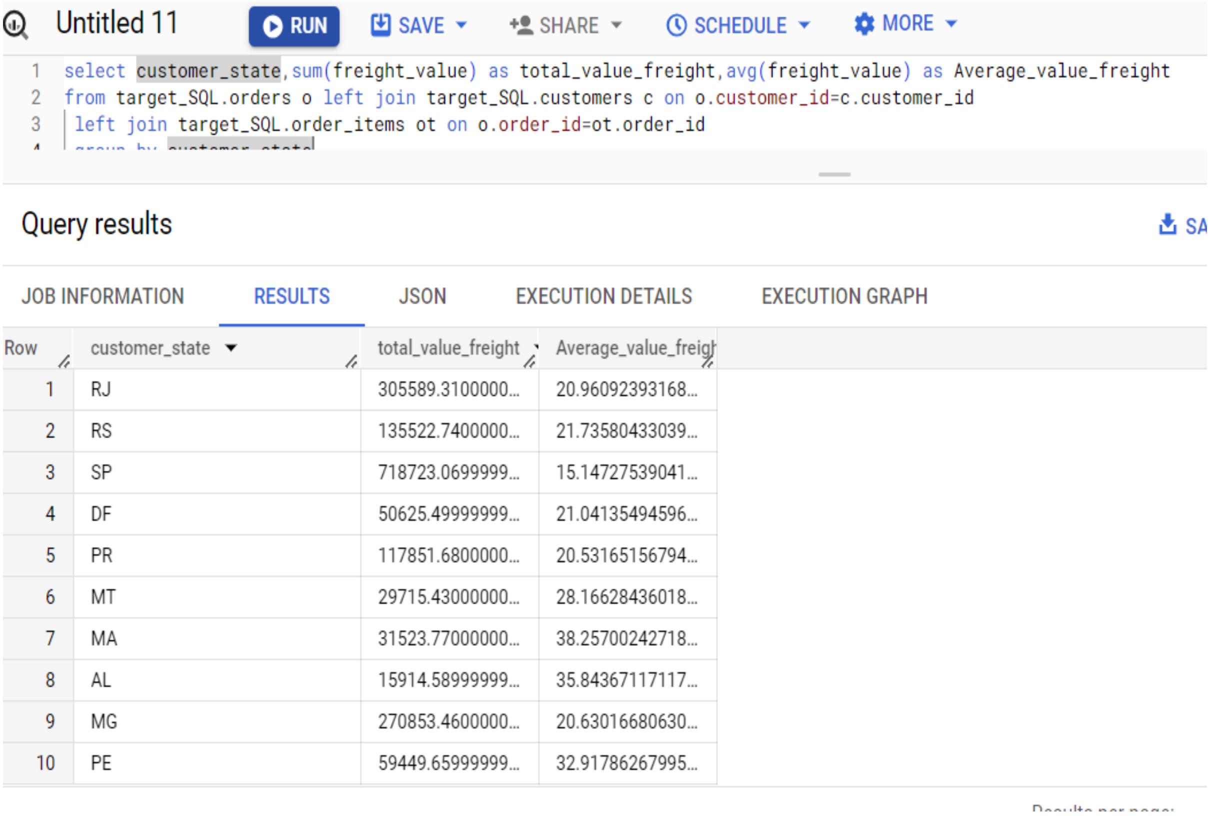
Insights:

- From the above results, we can see that SP is at the top in terms of total value of order price
- Whereas in terms of average value of order price it is not at top, which means most of the customers are buying from SP but the order amount per order is less

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

Query:

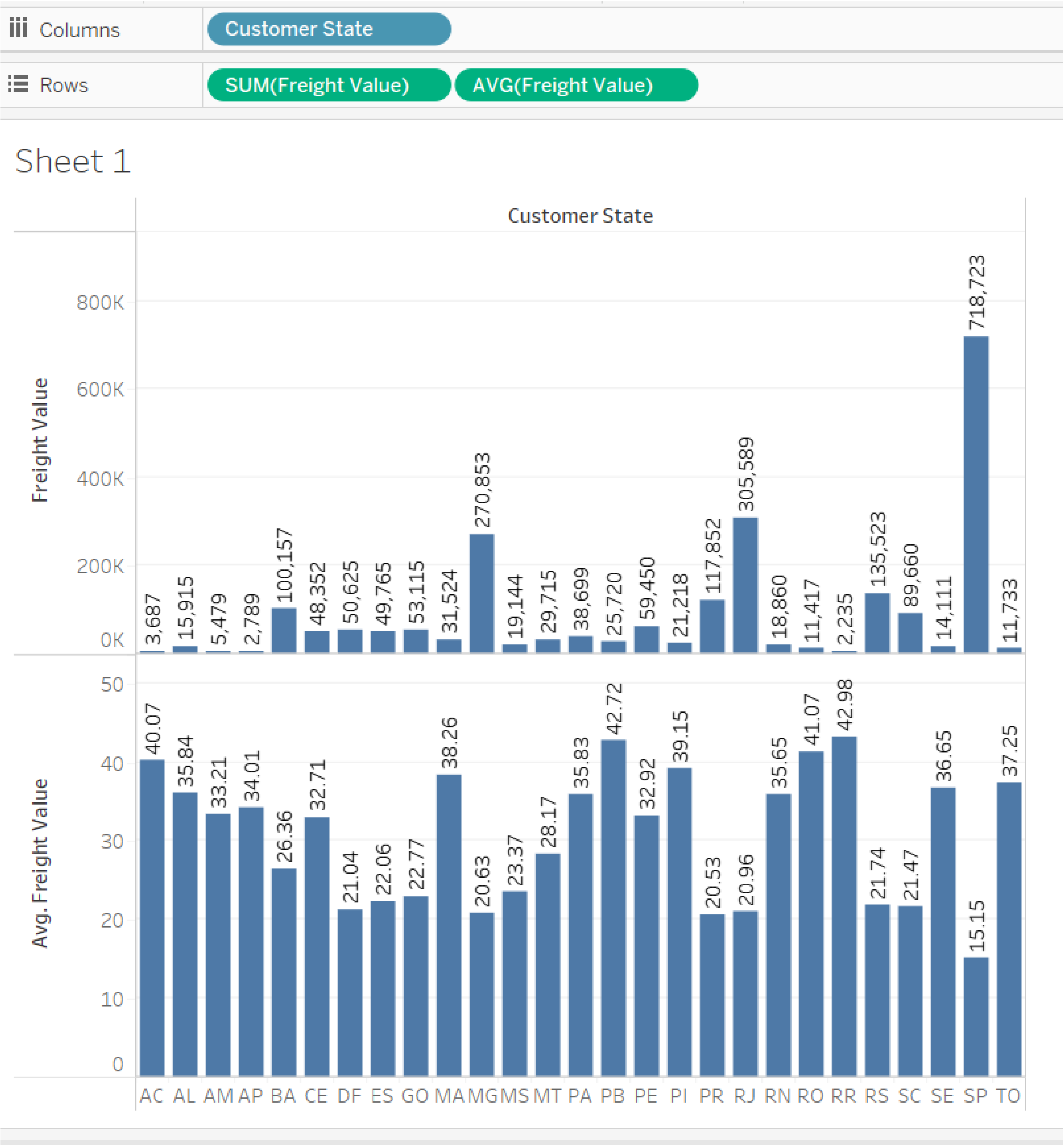
```
select customer_state,sum(freight_value) as total_value_freight,avg(freight_value) as

Average_value_freight from target_SQL.orders o left join target_SQL.customers c on

 o.customer_id=c.customer_id left join target_SQL.order_items ot on

o.order_id=ot.order_id group by customer_state
```



```
select customer_state,sum(freight_value) as total_value_freight,avg(freight_value) as Average_value_freight
from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id
 left join target_SQL.order_items ot on o.order_id=ot.order_id
 group by customer_state
```

Query results

Row	customer_state	total_value_freight	Average_value_freig
1	RJ	305589.3100000...	20.96092393168...
2	RS	135522.7400000...	21.73580433039...
3	SP	718723.0699999...	15.14727539041...
4	DF	50625.49999999...	21.04135494596...
5	PR	117851.6800000...	20.53165156794...
6	MT	29715.43000000...	28.16628436018...
7	MA	31523.77000000...	38.25700242718...
8	AL	15914.58999999...	35.84367117117...
9	MG	270853.4600000...	20.63016680630...
10	PE	59449.65999999...	32.91786267995...



Columns: Customer State

Rows: SUM(Freight Value) AVG(Freight Value)

Sheet 1

Insights:

- From the above results , we can see that SP is at the top in terms of total value of order freight

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

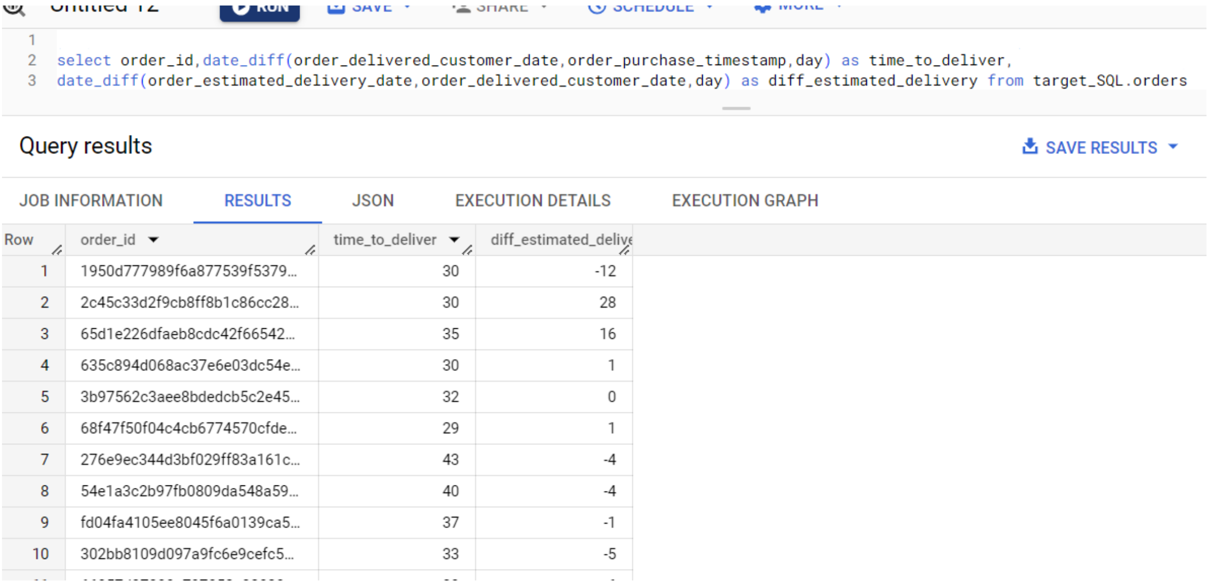
1.Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

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_estimated_delivery_date - order_delivered_customer_date

Query:

```
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 target_SQL.orders
```



```
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 target_SQL.orders
```

Query results

Row	order_id	time_to_deliver	diff_estimated_delive
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

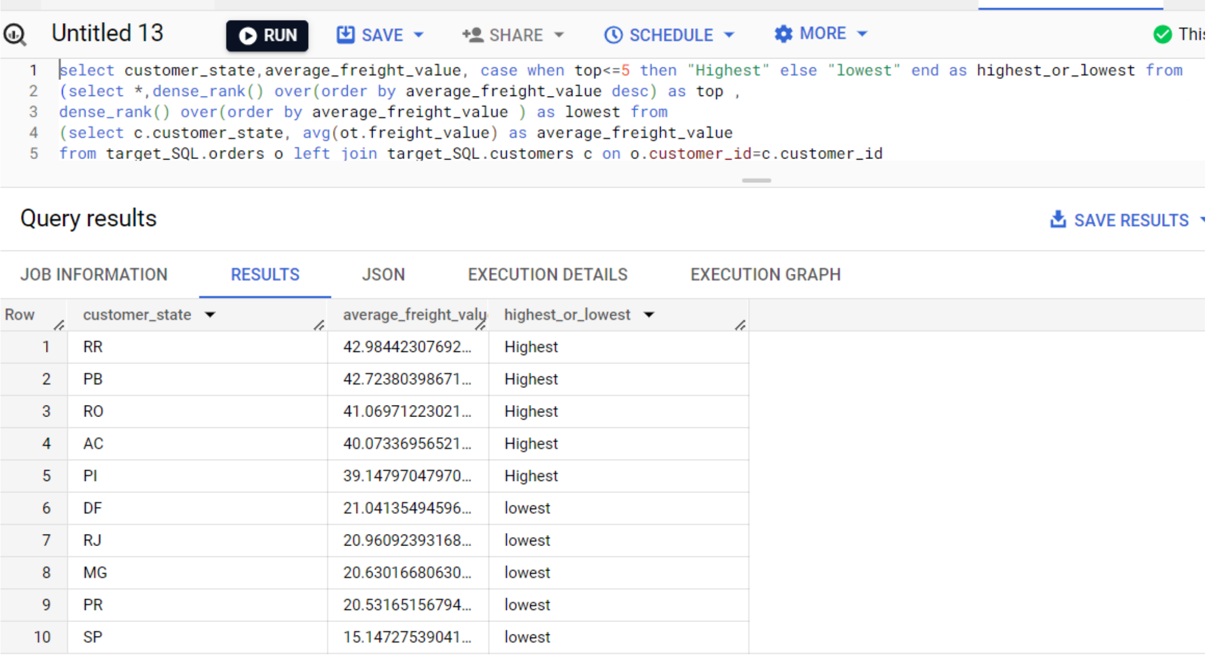
Insights:

- From the above results, we can see that in the diff_estimated_delivery column where ever there is negative value means that that the delivery was made after the estimated date.

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

Query:

```
select customer_state,average_freight_value, case when top<=5 then "Highest" else
"lowest" end as highest_or_lowest from (select *,dense_rank() over(order by
average_freight_value desc) as top ,dense_rank() over(order by average_freight_value ) as
lowest from(select c.customer_state, avg(ot.freight_value) as average_freight_value
from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id
left join target_SQL.order_items ot on o.order_id=ot.order_id group by c.customer_state))
where top<=5 or lowest<=5 order by top
```



```
select customer_state,average_freight_value, case when top<=5 then "Highest" else "lowest" end as highest_or_lowest from
(select *,dense_rank() over(order by average_freight_value desc) as top ,
dense_rank() over(order by average_freight_value ) as lowest from
(select c.customer_state, avg(ot.freight_value) as average_freight_value
from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id
```

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

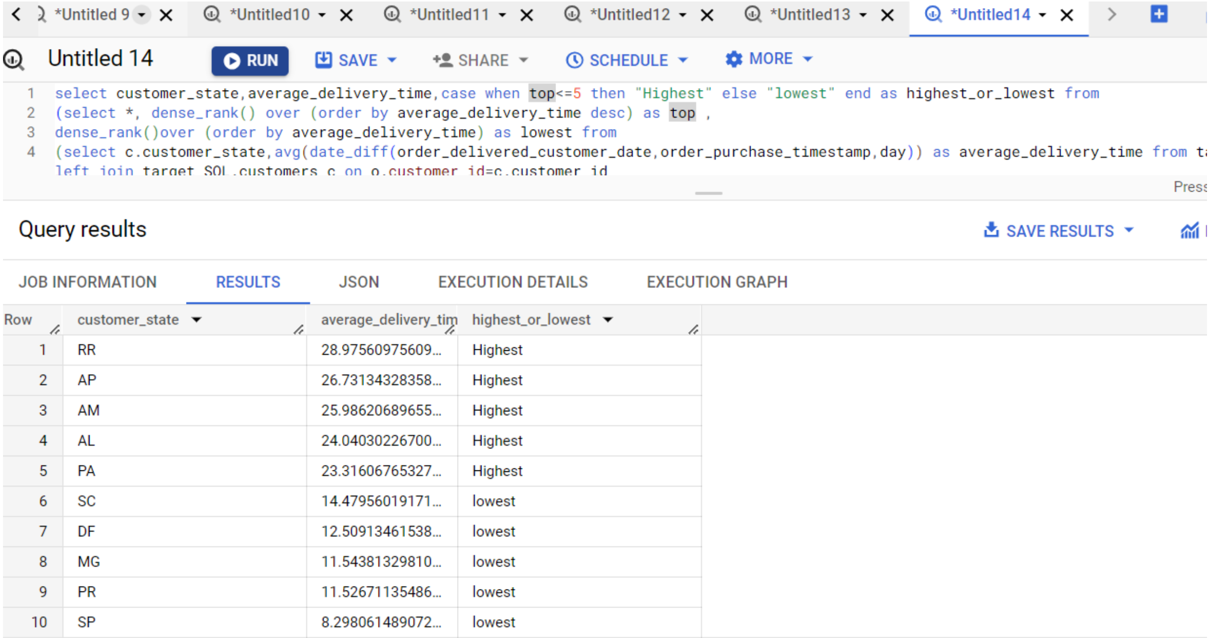
Insights:

- From the above results, we can say that RR,PB,RO,AC and PI are the top 5 states in terms of average freight value.
- SP,PR,MG,RJ and DF are the lowest 5 states in terms of average freight value.

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

Query:

select customer_state,average_delivery_time,case when top<=5 then "Highest" else "lowest" end as highest_or_lowest from (select *, dense_rank() over (order by average_delivery_time desc) as top ,dense_rank()over (order by average_delivery_time) as lowest from(select c.customer_state,avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,day)) as average_delivery_time from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id group by c.customer_state)) where top<=5 or lowest<=5 order by top



```
select customer_state,average_delivery_time,case when top<=5 then "Highest" else "lowest" end as highest_or_lowest from
(select *, dense_rank() over (order by average_delivery_time desc) as top ,
dense_rank()over (order by average_delivery_time) as lowest from
(select c.customer_state,avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)) as average_delivery_time from ta
left join target_SQL.customers c on o.customer_id=c.customer_id
```

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

Insights:

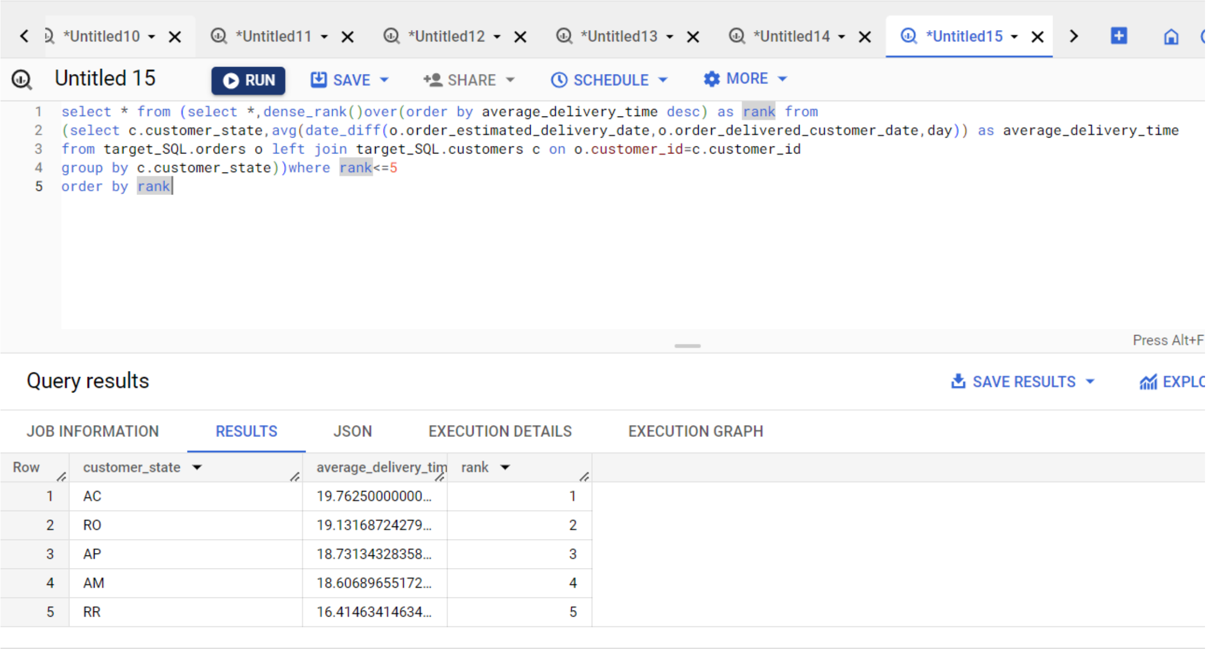
- From the above results, we can say that RR,AP,AM,AL and PA are the top 5 states in terms of average delivery time.
- SC,DF,MG,PR and SP are the lowest 5 states in terms of average delivery time.

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.

Query:

```
select * from (select *,dense_rank()over(order by average_delivery_time desc) as rank from

(select c.customer_state,avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)) as average_delivery_time from target_SQL.orders o left

join target_SQL.customers c on o.customer_id=c.customer_id  group by

c.customer_state))where rank<=5 order by rank
```



```
select * from (select *,dense_rank()over(order by average_delivery_time desc) as rank from
(select c.customer_state,avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)) as average_delivery_time
from target_SQL.orders o left join target_SQL.customers c on o.customer_id=c.customer_id
group by c.customer_state))where rank<=5
order by rank
```

Row	customer_state	average_delivery_tim	rank
1	AC	19.76250000000...	1
2	RO	19.13168724279...	2
3	AP	18.73134328358...	3
4	AM	18.60689655172...	4
5	RR	16.41463414634...	5

Insights:

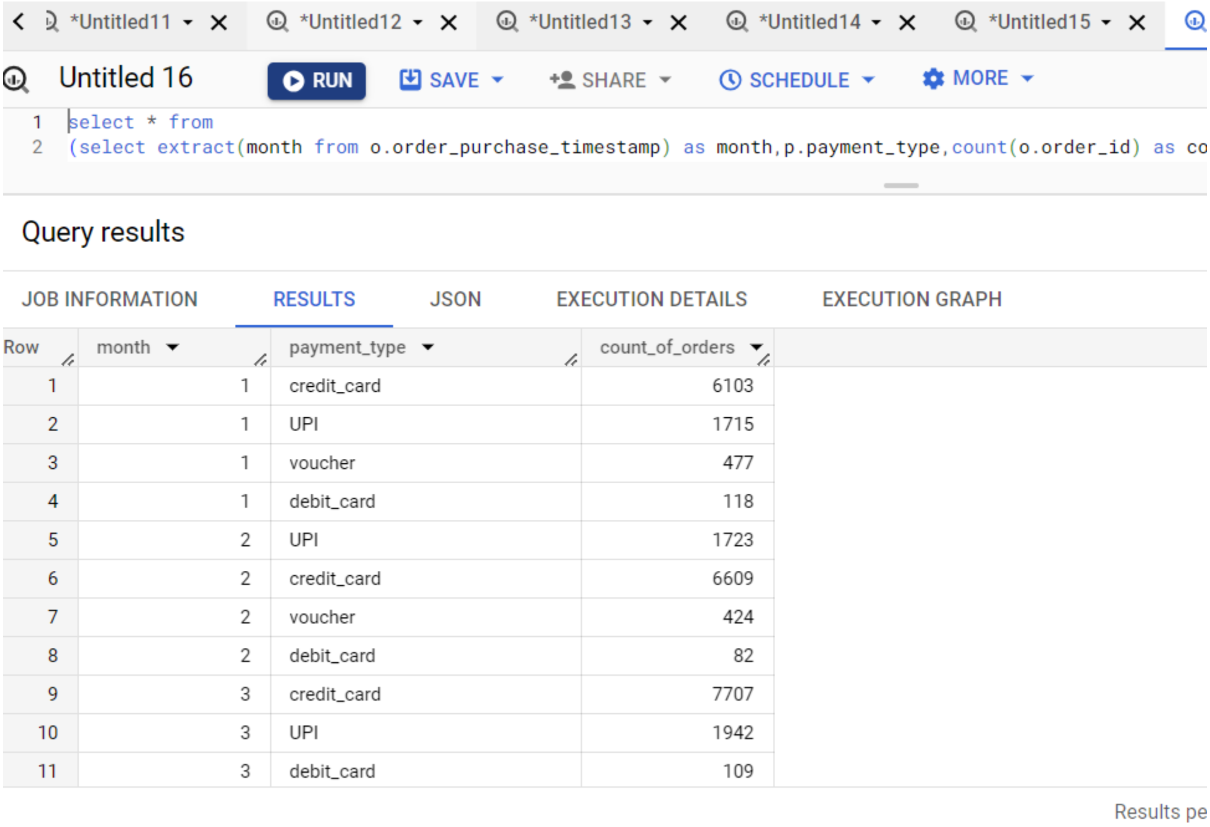
- AC,RO,AP,AM and RR are the states where delivery is really fast as the difference between the estimated delivery date and actual delivery date is more.

6.Analysis based on the payments:

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

Query:

select * from (select extract(month from o.order_purchase_timestamp) as month,p.payment_type,count(o.order_id) as count_of_orders from target_SQL.orders o left join target_SQL.payments p on o.order_id=p.order_id group by extract(month from o.order_purchase_timestamp),p.payment_type) order by month



```
select * from
(select extract(month from o.order_purchase_timestamp) as month,p.payment_type,count(o.order_id) as co
```

Row	month	payment_type	count_of_orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6609
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942
11	3	debit_card	109

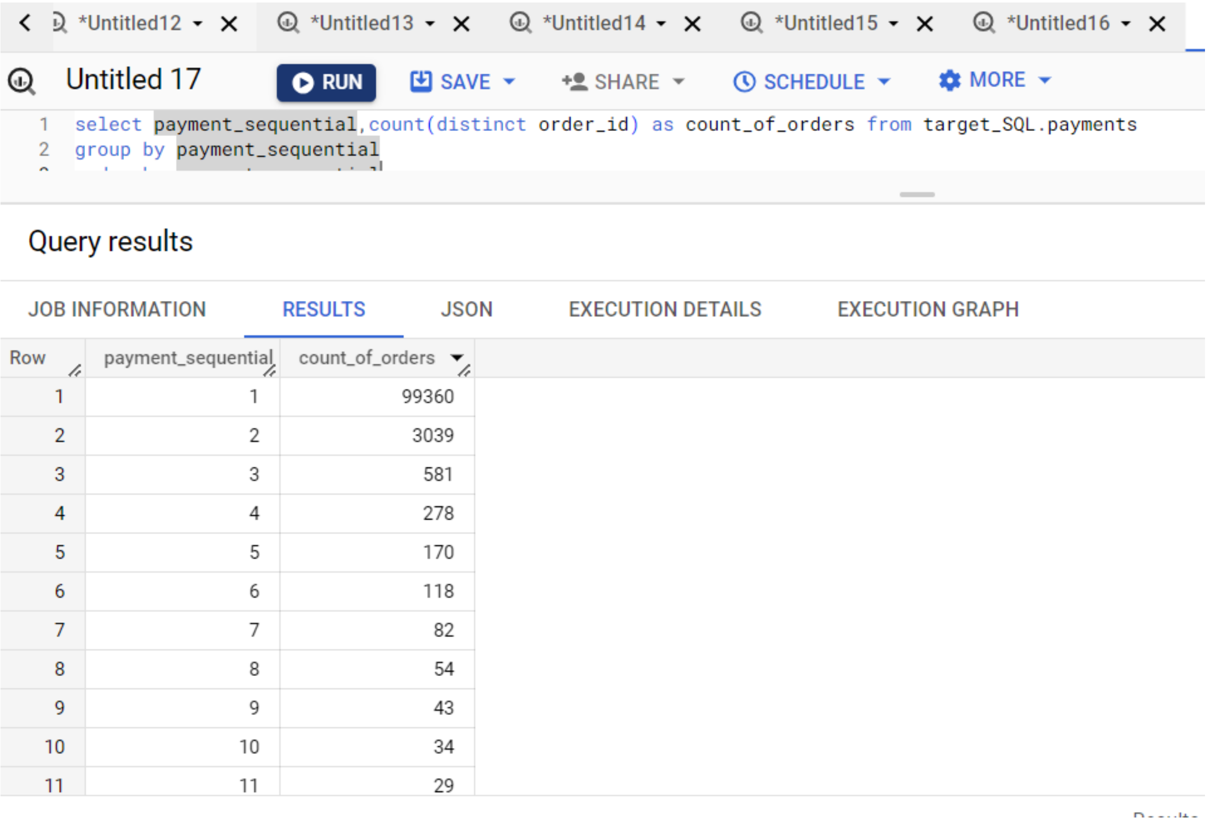
Insights:

- From the above results , we can see that maximum payments are made through credit card over each month.

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

Query:

```
select payment_sequential,count(distinct order_id) as count_of_orders from

target_SQL.payments group by payment_sequential order by payment_sequential
```



```
select payment_sequential,count(distinct order_id) as count_of_orders from target_SQL.payments
group by payment_sequential
```

Row	payment_sequential	count_of_orders
1	1	99360
2	2	3039
3	3	581
4	4	278
5	5	170
6	6	118
7	7	82
8	8	54
9	9	43
10	10	34
11	11	29

Insights:

- From the above results, we can see that most of the orders has paid the first instalment.

Recommendations:

- This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. This involves around 8011 cities and 27 states in Brazil. There is an upward trend seen in the number of orders placed from 2016 to 2018 and most of the orders were placed in the month of May, July and August and least number of orders were placed in the month of September, October and December. During these months where the number of orders is low we can provide incentives such as discounts, special promotions, loyalty programs, or referral rewards to attract new customers and encourage repeat business.

- Maximum of the customers are from SP. To increase the customers involvement in other states, we need to build a strong brand identity. This can be achieved by digital marketing (websites, social media, email campaigns, search engine optimization) or by traditional marketing like posters, radio, tv advertisement.

- Need to provide exceptional customer services by addressing customer inquiries, concerns, and complaints to built trust and loyalty. This will help in word of mouth marketing.

- To increase the reach, we can build strategic partnerships like collaborating with influencers.