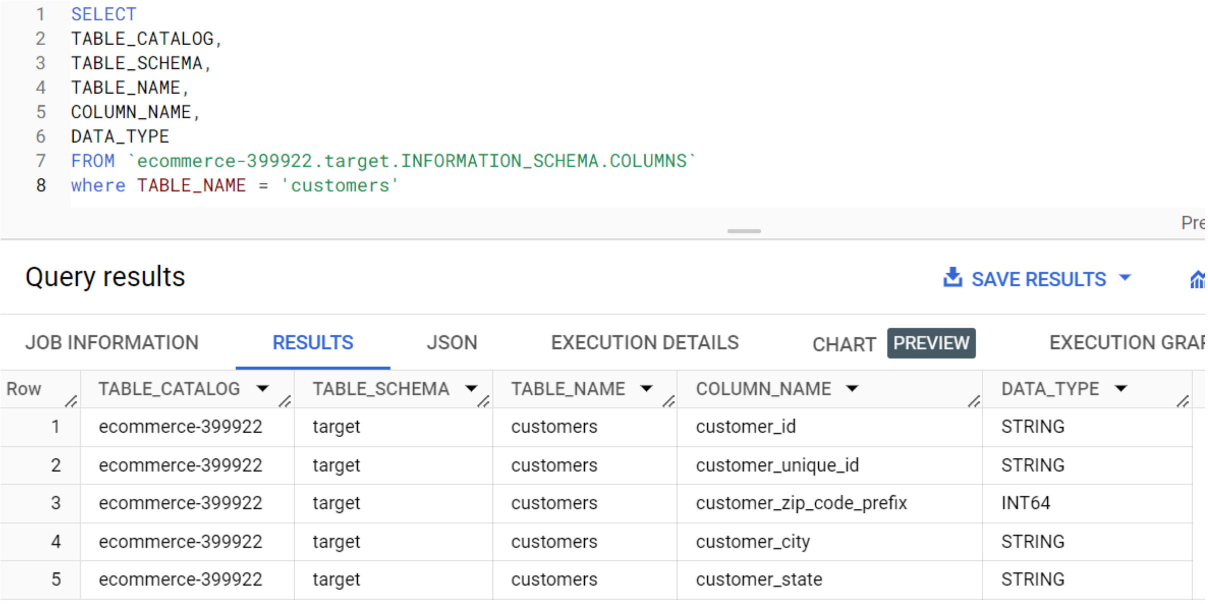
Target business case Study

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

A. Data type of all columns in the “customers” table.

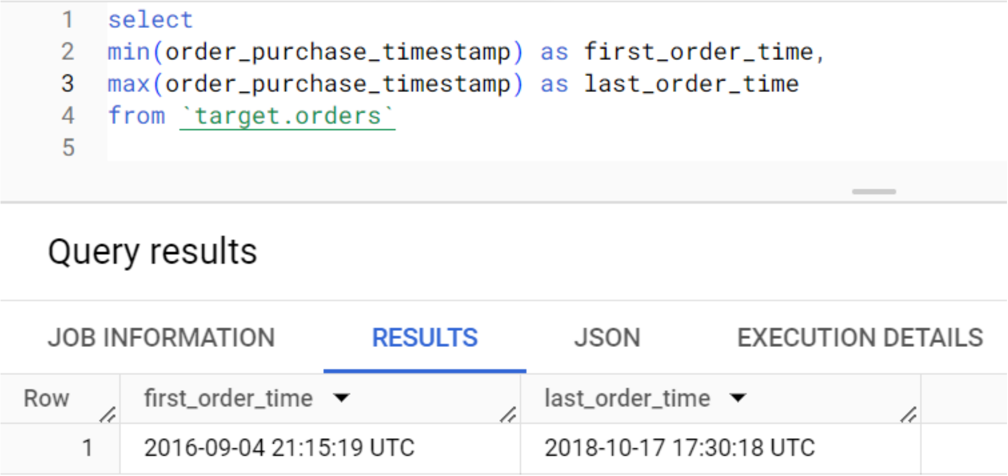


```
SELECT
TABLE_CATALOG,
TABLE_SCHEMA,
TABLE_NAME,
COLUMN_NAME,
DATA_TYPE
FROM `ecommerce-399922.target.INFORMATION_SCHEMA.COLUMNS`
where TABLE_NAME = 'customers'
```

Row	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
1	ecommerce-399922	target	customers	customer_id	STRING
2	ecommerce-399922	target	customers	customer_unique_id	STRING
3	ecommerce-399922	target	customers	customer_zip_code_prefix	INT64
4	ecommerce-399922	target	customers	customer_city	STRING
5	ecommerce-399922	target	customers	customer_state	STRING

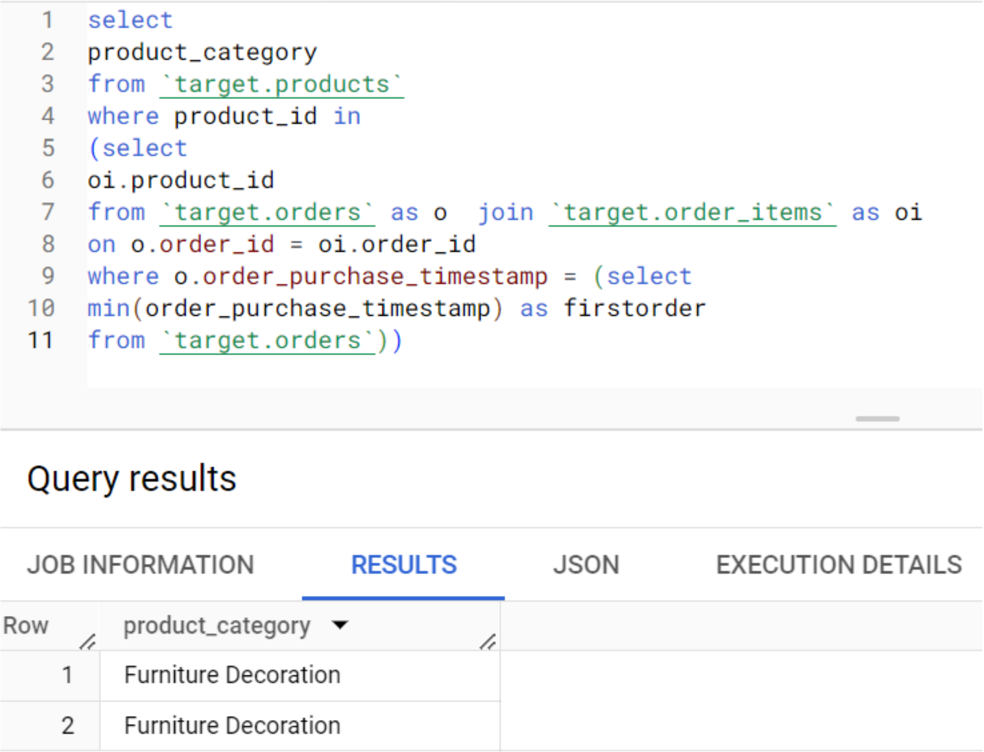
Insight: Target’s Customer table contains 5 different columns, except Zip-code column rest all String data type. It is for learning how to Data types of columns from a table.

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



```
select
min(order_purchase_timestamp) as first_order_time,
max(order_purchase_timestamp) as last_order_time
from `target.orders`

```

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



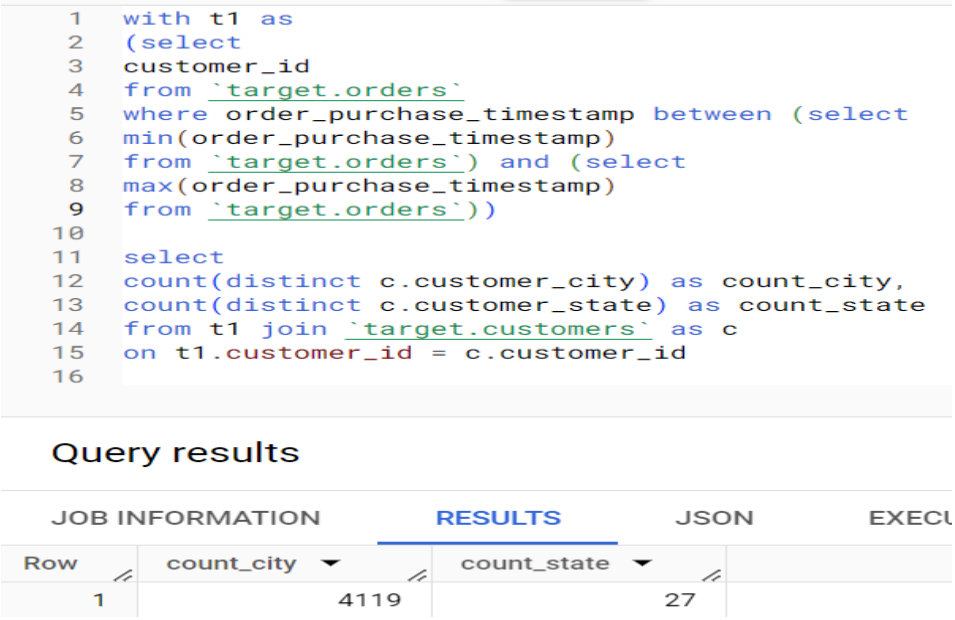
```sql
select
product_category
from `target.products`
where product_id in
(select
oi.product_id
from `target.orders` as o  join `target.order_items` as oi
on o.order_id = oi.order_id
where o.order_purchase_timestamp = (select
min(order_purchase_timestamp) as firstorder
from `target.orders`))
```

Row	product_category
1	Furniture Decoration
2	Furniture Decoration

Insight: As per the data available from Target business case, the first order was on 4th of September, 2016 from the city called Boa Vista, RR state and bought 2 furniture decoration items with total payment of

And the last order was on 17th October 2018 from the city called Sorocaba, SP state.

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



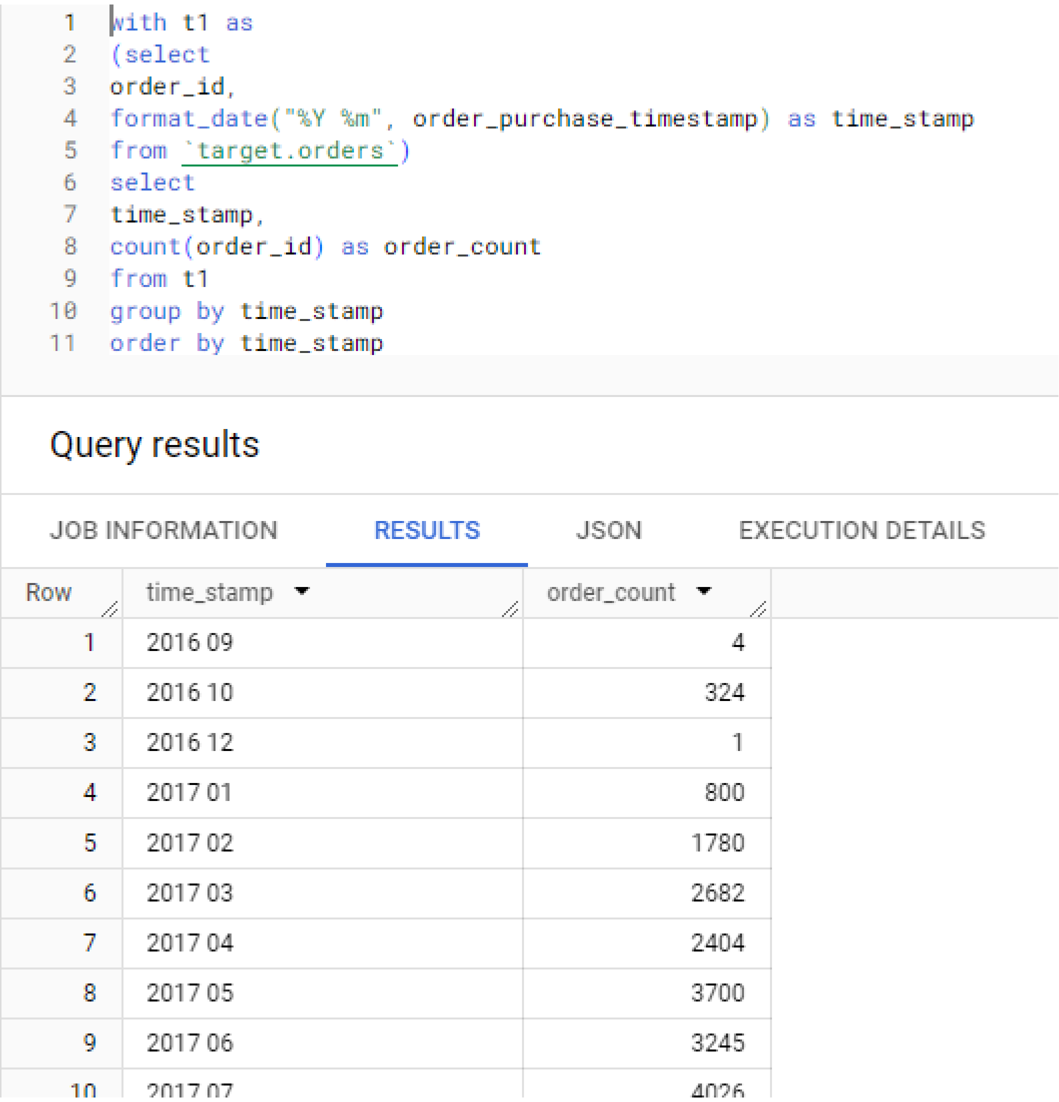
```sql
with t1 as
(select
customer_id
from `target.orders`
where order_purchase_timestamp between (select
min(order_purchase_timestamp)
from `target.orders`) and (select
max(order_purchase_timestamp)
from `target.orders`))

select
count(distinct c.customer_city) as count_city,
count(distinct c.customer_state) as count_state
from t1 join `target.customers` as c
on t1.customer_id = c.customer_id
```

Row	count_city	count_state
1	4119	27

Insight: From order date to last order date duration, during this 26 months of duration the customers spread across 27 states and 4119 cities.

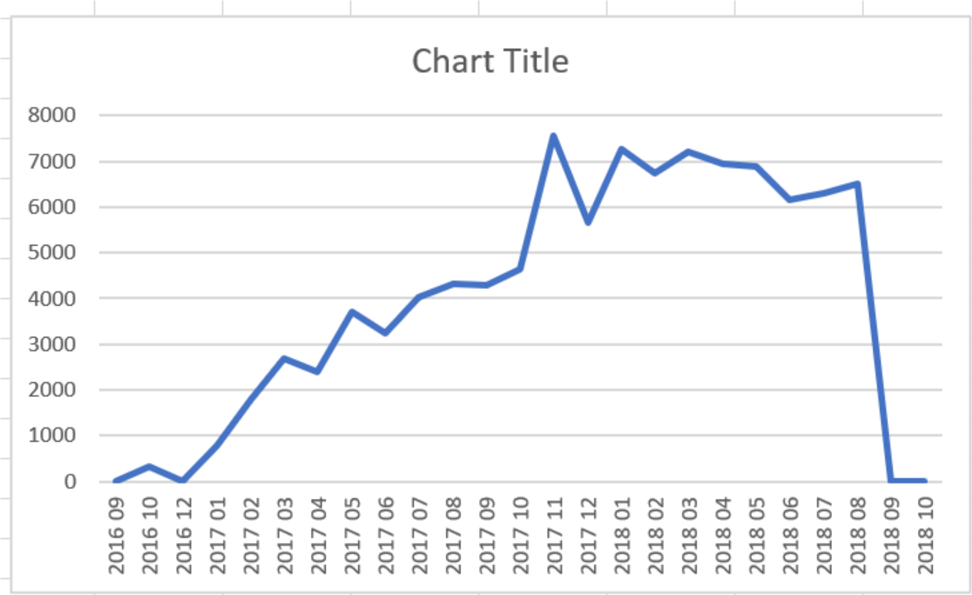
II In-depth Exploration:

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



```
with t1 as
(select
order_id,
format_date("%Y %m", order_purchase_timestamp) as time_stamp
from `target.orders`)
select
time_stamp,
count(order_id) as order_count
from t1
group by time_stamp
order by time_stamp
```

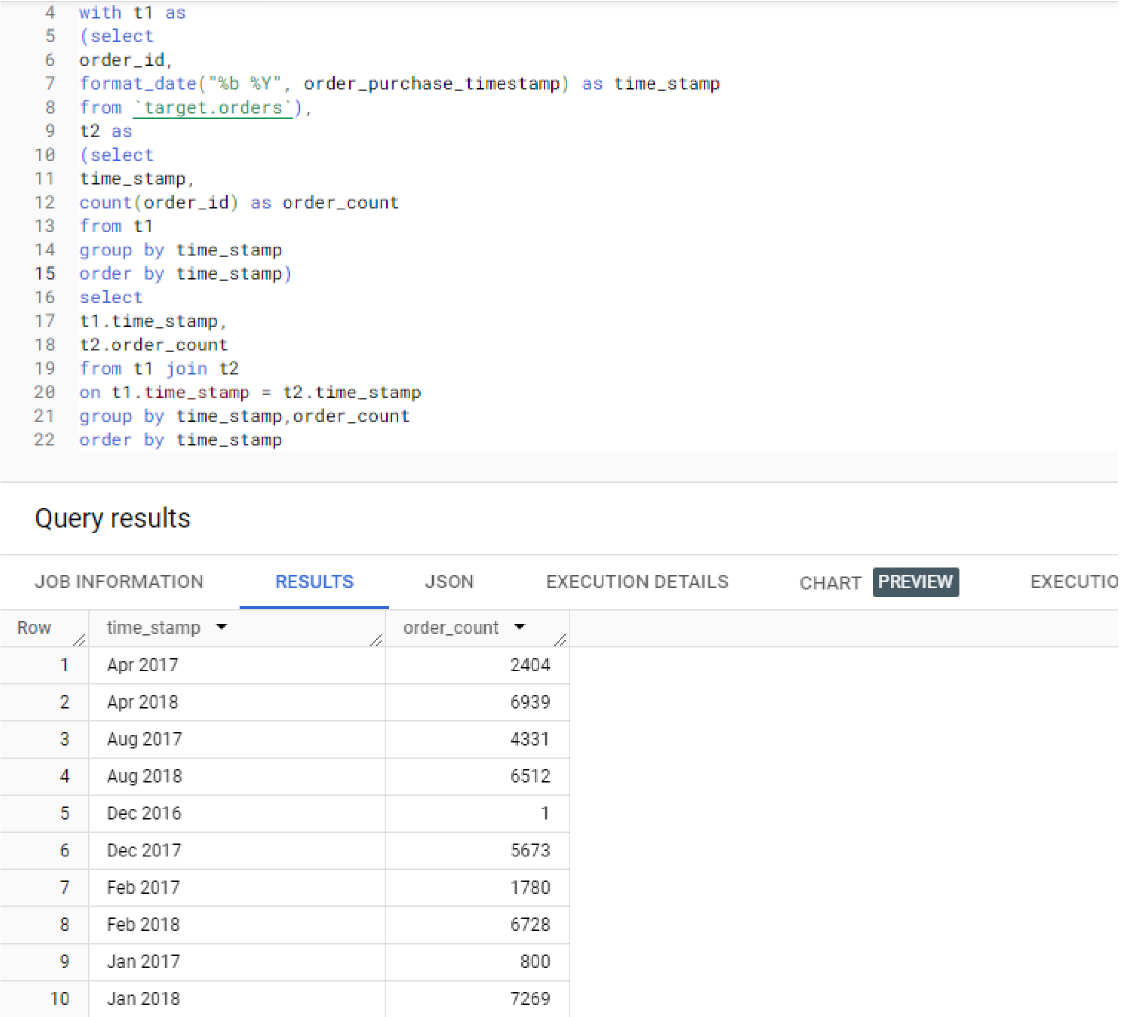
Row	time_stamp	order_count
1	2016 09	4
2	2016 10	324
3	2016 12	1
4	2017 01	800
5	2017 02	1780
6	2017 03	2682
7	2017 04	2404
8	2017 05	3700
9	2017 06	3245
10	2017 07	4026



From the chart, it's evident that there is a general upward trend in the number of orders placed from late 2016 through 2017. However, there seems to be a decline in late 2018.

In summary, there was a growing trend in the number of orders placed from 2016 through most of 2017, but the trend started to decline in late 2018.

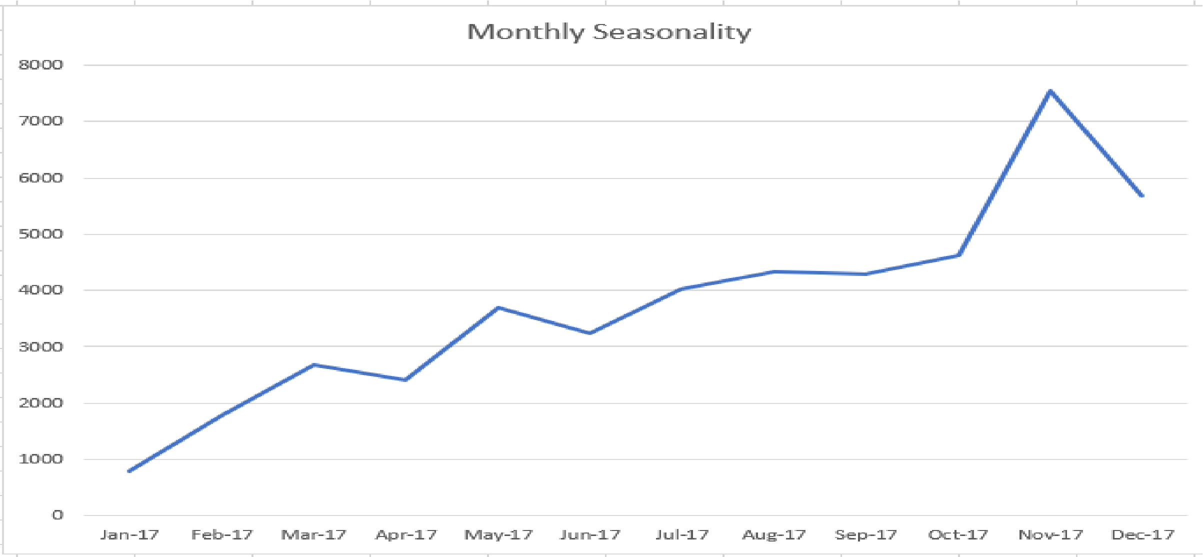
B. Can we see some kind of monthly seasonality in terms of the no.of orders being placed?



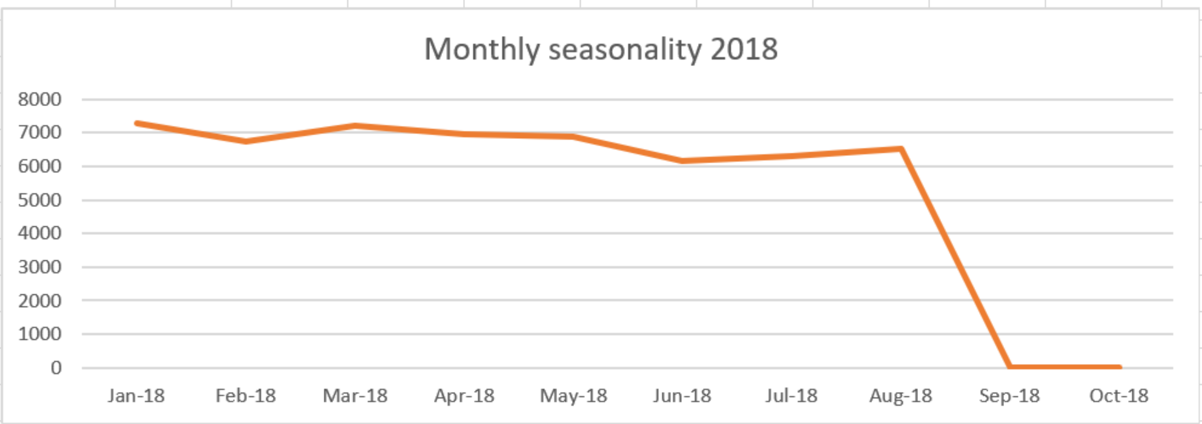
```
with t1 as
(select
order_id,
format_date("%b %Y", order_purchase_timestamp) as time_stamp
from `target.orders`),
t2 as
(select
time_stamp,
count(order_id) as order_count
from t1
group by time_stamp
order by time_stamp)
select
t1.time_stamp,
t2.order_count
from t1 join t2
on t1.time_stamp = t2.time_stamp
group by time_stamp,order_count
order by time_stamp
```

Query results

Row	time_stamp	order_count
1	Apr 2017	2404
2	Apr 2018	6939
3	Aug 2017	4331
4	Aug 2018	6512
5	Dec 2016	1
6	Dec 2017	5673
7	Feb 2017	1780
8	Feb 2018	6728
9	Jan 2017	800
10	Jan 2018	7269



March and May have higher order counts compared to the surrounding months. This could be attributed to various factors such as promotions, special events, or season-specific trends. While November and December clearly stand out as the highest order count months due to holiday shopping, March and May have their own spikes.



While the first eight months of 2018 show some variability, there is a substantial decrease in order counts in September and October 2018 based on the provided data. Analysing the causes of these decreases and identifying strategies to address them would be important for the business.

Summary- 2017:

The year 2017 started with relatively low order counts in January and February (800 and 1780, respectively).

Order counts began to increase significantly from March through August, with the highest peak in November (7544) and a strong finish in December (5673), likely driven by holiday shopping.

Overall, 2017 exhibited a clear seasonal trend, with order counts increasing towards the end of the year.

2018:

2018 started strong with high order counts in January (7269) and February (6728).

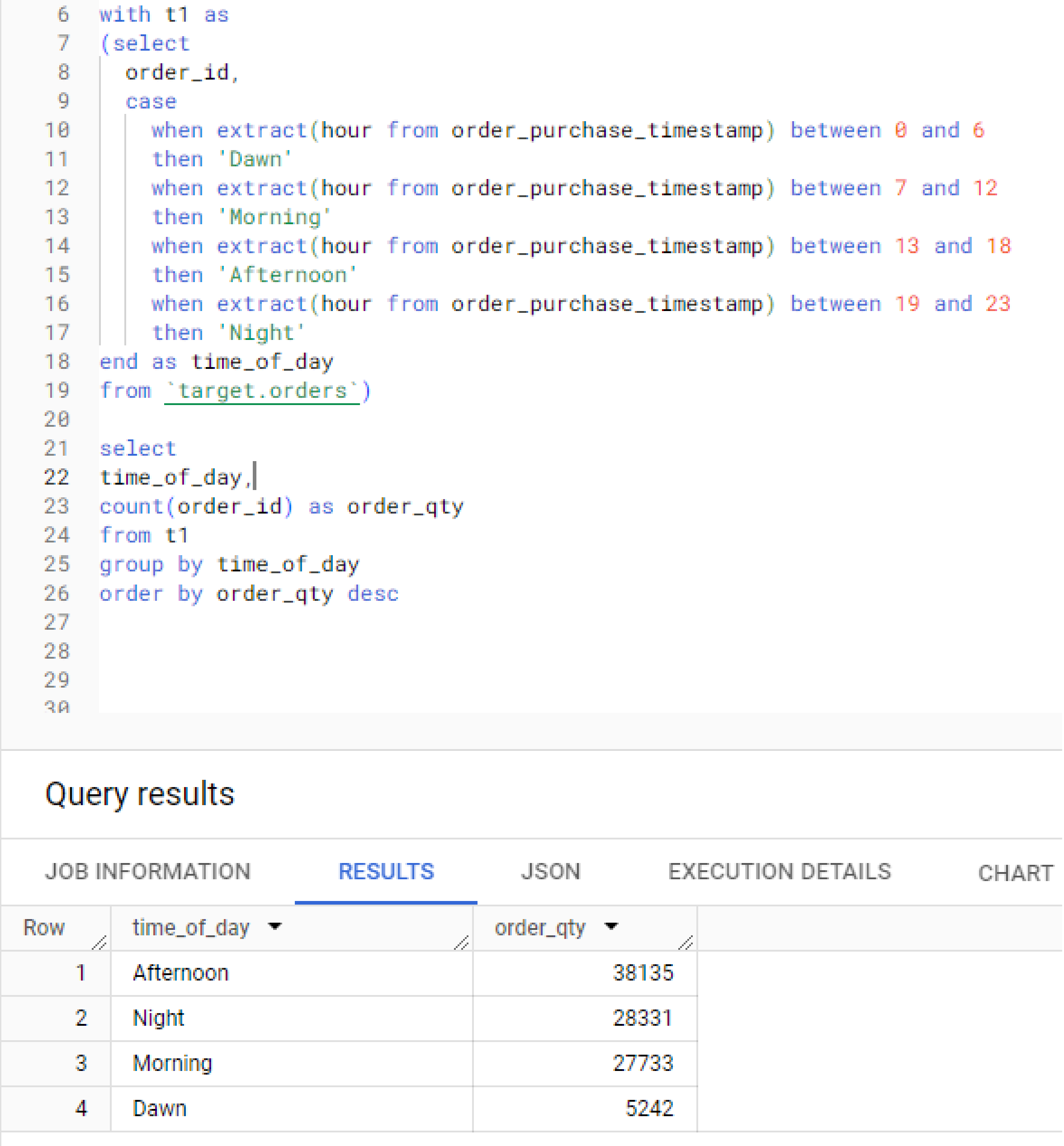
However, there was a noticeable drop in order counts in the last two months of the year (September and October) with just 10 orders in September and October combined.

Unlike 2017, 2018 did not follow a clear seasonal trend throughout the year. Instead, it started well but experienced a significant decline in the latter part of the year.

In summary, the data suggests that the business experienced a seasonal trend in 2017, with increasing order counts towards the end of the year, likely due to holiday-related shopping. However, in 2018, there was a strong start to the year, but the business faced challenges in maintaining order counts in the latter part of the year, with a notable decline in September and October.

It's important for the business to understand the factors contributing to these trends, especially the drop in orders in late 2018, and consider strategies to address any challenges in order to achieve consistent growth and maintain customer engagement throughout the year. Additional data and context would be valuable for a more comprehensive analysis and informed decision-making.

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



```
with t1 as
(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 '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 time_of_day
from `target.orders`)

select
time_of_day,
count(order_id) as order_qty
from t1
group by time_of_day
order by order_qty desc
```

Row	time_of_day	order_qty
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

Order Quantity Distribution: The table provides information on the distribution of order quantities across different times of the day.

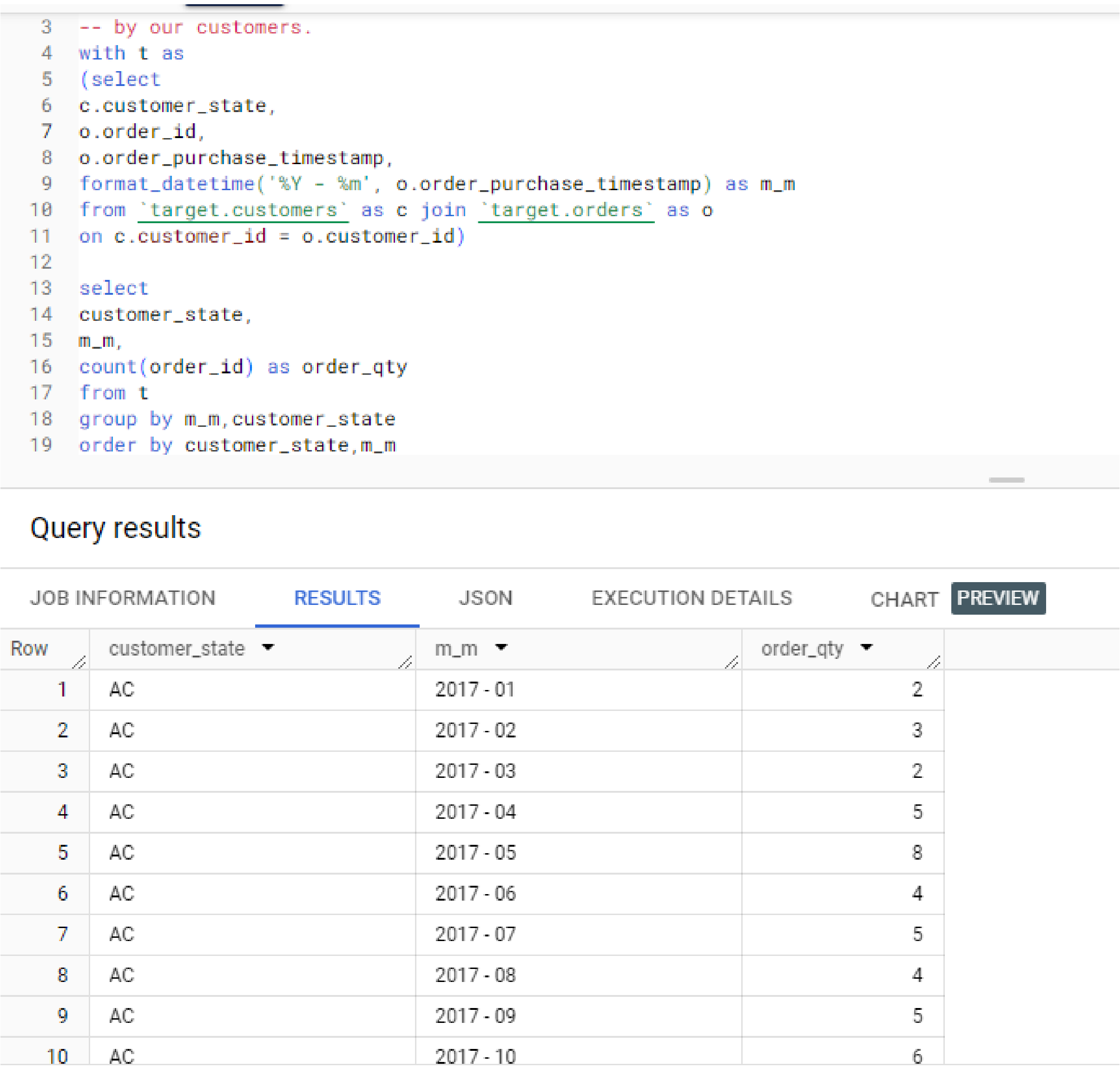
Peak Ordering Time: The "Afternoon" appears to be the peak ordering time, with the highest order quantity of 38,135. This suggests that a significant portion of orders is placed during the afternoon hours.

Night and Morning Orders: The "Night" and "Morning" times also show substantial order quantities of 28,331 and 27,733, respectively. This indicates that orders are fairly evenly distributed between night time and morning hours.

Dawn Orders: The "Dawn" time has the lowest order quantity, with 5,242 orders. This suggests that fewer orders are placed during the early morning hours.

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

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



```
-- by our customers.
with t as
(select
c.customer_state,
o.order_id,
o.order_purchase_timestamp,
format_datetime('%Y - %m', o.order_purchase_timestamp) as m_m
from `target.customers` as c join `target.orders` as o
on c.customer_id = o.customer_id)

select
customer_state,
m_m,
count(order_id) as order_qty
from t
group by m_m,customer_state
order by customer_state,m_m
```

Query results

Row	customer_state	m_m	order_qty
1	AC	2017 - 01	2
2	AC	2017 - 02	3
3	AC	2017 - 03	2
4	AC	2017 - 04	5
5	AC	2017 - 05	8
6	AC	2017 - 06	4
7	AC	2017 - 07	5
8	AC	2017 - 08	4
9	AC	2017 - 09	5
10	AC	2017 - 10	6

Insights:

There is a wide variation in order quantities between states and over time. Some states consistently have high order quantities, while others show fluctuations.

High-Performing States:

MG),(SP), and (RJ) consistently have high order quantities throughout the entire period.

Seasonal Patterns:

Some states exhibit clear seasonal patterns in their order quantities. For example, in the state of BA), order quantities peak around the middle of the year (May to July), which might coincide with specific seasons or holidays.

Steady Growth:

Some states, such as (ES) and (SC), show relatively steady growth in order quantities over time. This indicates a consistent increase in demand.

States with Fluctuations:

States like (AP), (RR), and (AC) exhibit fluctuations in their order quantities, suggesting less predictable demand patterns.

Significant Increases:

Some states, like (PB), PI), and (TO), experience significant increases in order quantities over the given period.

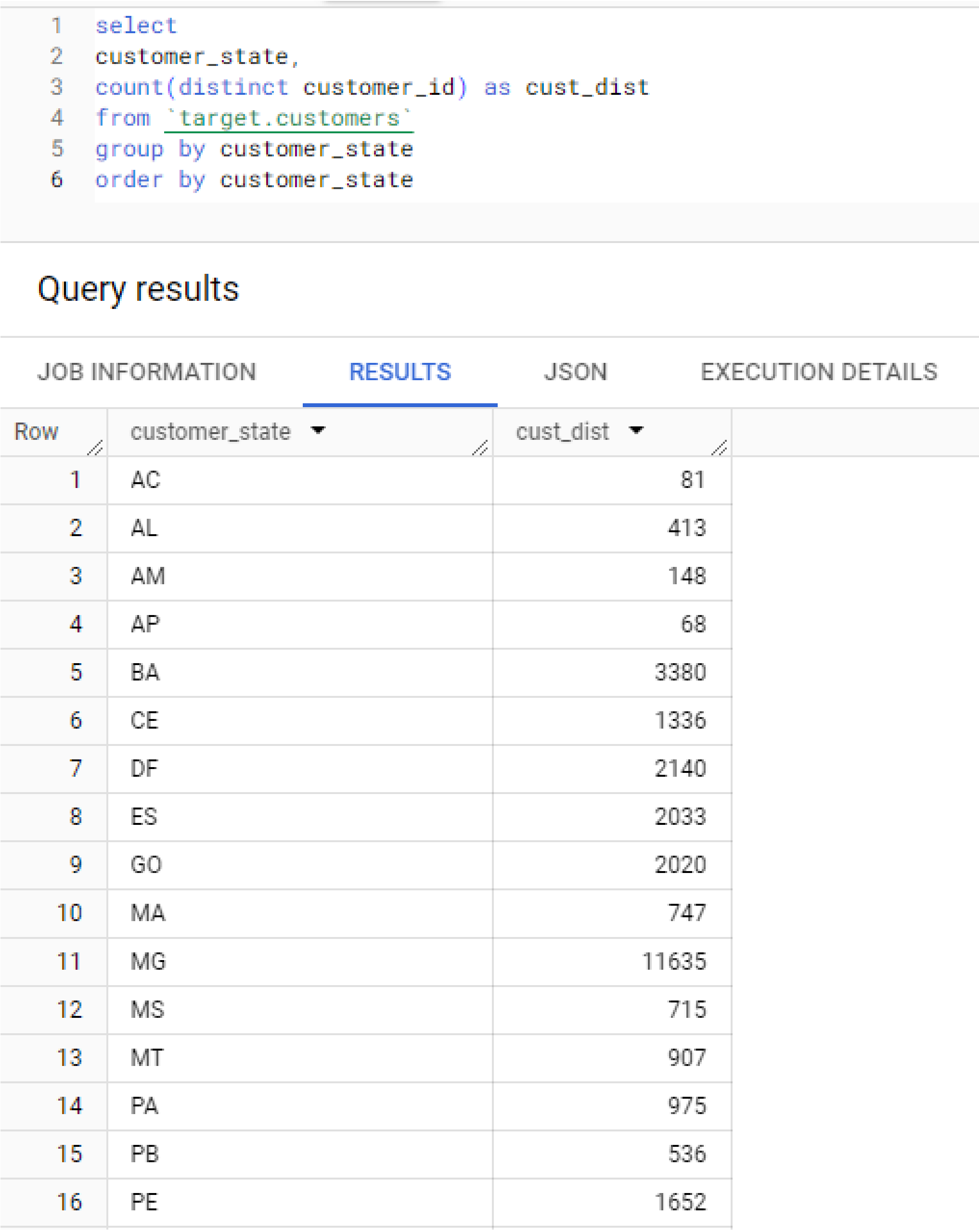
States with Decreasing Trends:

A few states, like (RO), seem to have a declining trend in order quantities. This might warrant further investigation into the underlying causes.

Outliers:

There are some outliers in the data, where certain months or states show exceptionally high or low order quantities compared to their usual patterns. These anomalies should be examined more closely.

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



```
select
customer_state,
count(distinct customer_id) as cust_dist
from `target.customers`
group by customer_state
order by customer_state
```

Query results

Row	customer_state	cust_dist
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635
12	MS	715
13	MT	907
14	PA	975
15	PB	536
16	PE	1652

Highest Customer Count: (SP) has the highest number of customers with 41,746 customers, indicating a significant customer base in this state.

Lowest Customer Counts: (RR) has the lowest number of customers with only 46 customers, indicating a relatively smaller customer base.

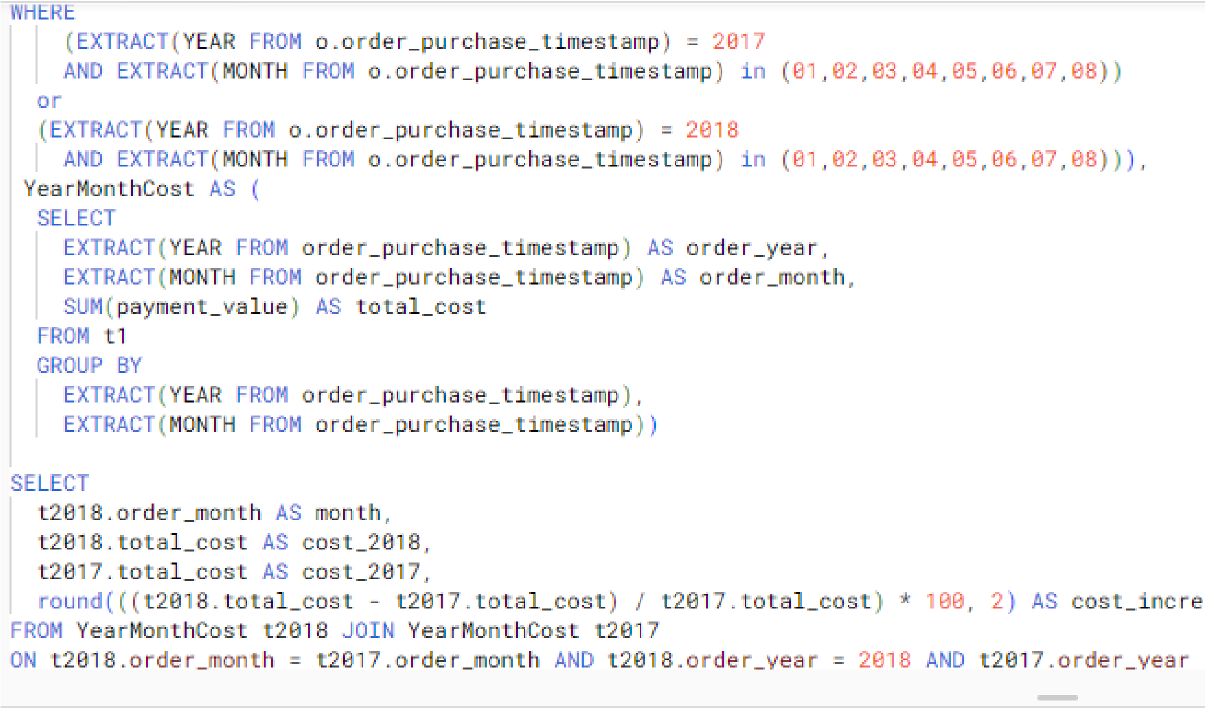
Total Customer Count: The total number of customers across all states is 84,785.

Variation Across States: The customer distribution data shows that customer counts can vary significantly from one state to another, reflecting differences in population and economic activity.

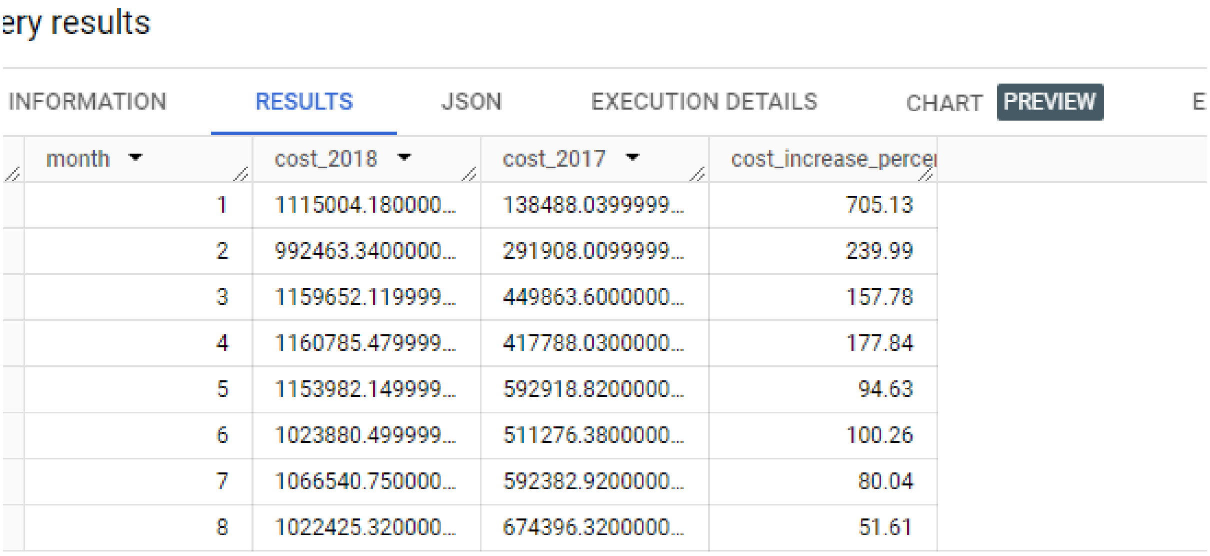
Significant Customer Counts: (MG) and (RJ) also have substantial customer counts with 11,635 and 12,852 customers, respectively. These states have a strong customer presence.

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

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



```
WHERE
    (EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) in (01,02,03,04,05,06,07,08))
  or
  (EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) in (01,02,03,04,05,06,07,08))),
 YearMonthCost AS (
  SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
    SUM(payment_value) AS total_cost
  FROM t1
  GROUP BY
    EXTRACT(YEAR FROM order_purchase_timestamp),
    EXTRACT(MONTH FROM order_purchase_timestamp))

SELECT
  t2018.order_month AS month,
  t2018.total_cost AS cost_2018,
  t2017.total_cost AS cost_2017,
  round(((t2018.total_cost - t2017.total_cost) / t2017.total_cost) * 100, 2) AS cost_incre
FROM YearMonthCost t2018 JOIN YearMonthCost t2017
ON t2018.order_month = t2017.order_month AND t2018.order_year = 2018 AND t2017.order_year
```

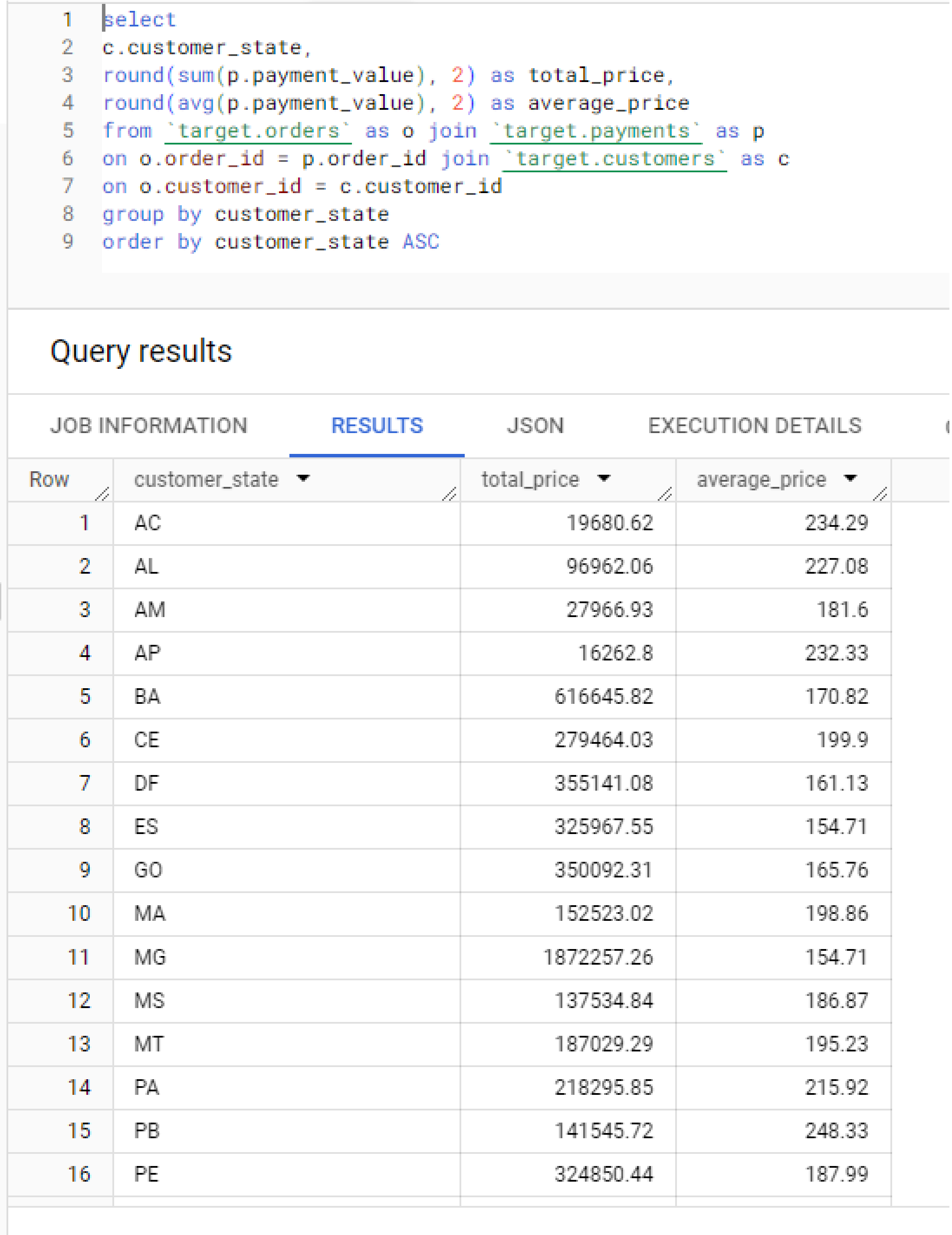


month	cost_2018	cost_2017	cost_increase_perce
1	1115004.180000...	138488.0399999...	705.13
2	992463.3400000...	291908.0099999...	239.99
3	1159652.119999...	449863.6000000...	157.78
4	1160785.479999...	417788.0300000...	177.84
5	1153982.149999...	592918.8200000...	94.63
6	1023880.499999...	511276.3800000...	100.26
7	1066540.750000...	592382.9200000...	80.04
8	1022425.320000...	674396.3200000...	51.61

Insight:

There is a significant growth in the month of January 2018 comparatively January 2017, and it been decreasing gradually to the month of August 2018. But overall there is 137% growth in cost of orders.

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



```
select
c.customer_state,
round(sum(p.payment_value), 2) as total_price,
round(avg(p.payment_value), 2) as average_price
from `target.orders` as o join `target.payments` as p
on o.order_id = p.order_id join `target.customers` as c
on o.customer_id = c.customer_id
group by customer_state
order by customer_state ASC
```

Row	customer_state	total_price	average_price
1	AC	19680.62	234.29
2	AL	96962.06	227.08
3	AM	27966.93	181.6
4	AP	16262.8	232.33
5	BA	616645.82	170.82
6	CE	279464.03	199.9
7	DF	355141.08	161.13
8	ES	325967.55	154.71
9	GO	350092.31	165.76
10	MA	152523.02	198.86
11	MG	1872257.26	154.71
12	MS	137534.84	186.87
13	MT	187029.29	195.23
14	PA	218295.85	215.92
15	PB	141545.72	248.33
16	PE	324850.44	187.99

Insight:

Highest Total Price: (SP) has the highest total price, amounting to approximately 5,998,226.96 units of currency. This is expected, given that São Paulo is one of the most populous and economically significant states in the dataset.

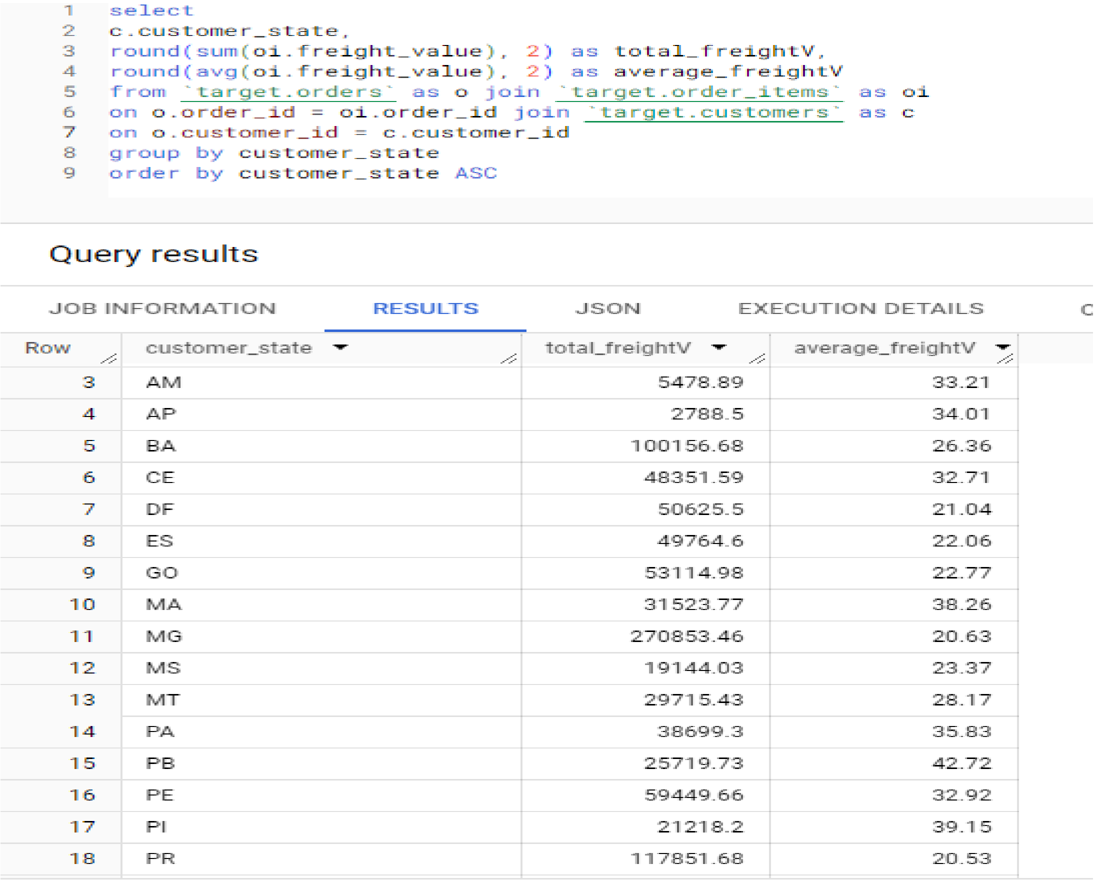
Lowest Total Price: (RR) has the lowest total price, with a value of approximately 10,064.62 units of currency. Roraima is one of the smaller states in terms of population and economic activity.

Average Price Variations: There are notable variations in average prices across states. (PB) has the highest average price, approximately 248.33 units of currency. In contrast, (RS) has a lower average price, around 157.18 units of currency.

Economic Variations: States like (SP), (RJ), and (MG) have high total prices, reflecting their significant economic activity and population. On the other hand, states like (RR) and (AP) have lower total prices, which may be attributed to their smaller populations and economic scale.

Average order Price Insights: The average price can provide insights into the pricing strategy and consumer behaviour in each state. States with higher average prices may indicate a market for premium or luxury products, while lower average prices might suggest more affordability or competition.

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



```
select
c.customer_state,
round(sum(oi.freight_value), 2) as total_freightV,
round(avg(oi.freight_value), 2) as average_freightV
from `target.orders` as o join `target.order_items` as oi
on o.order_id = oi.order_id join `target.customers` as c
on o.customer_id = c.customer_id
group by customer_state
order by customer_state ASC
```

Query results

Row	customer_state	total_freightV	average_freightV
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26
11	MG	270853.46	20.63
12	MS	19144.03	23.37
13	MT	29715.43	28.17
14	PA	38699.3	35.83
15	PB	25719.73	42.72
16	PE	59449.66	32.92
17	PI	21218.2	39.15
18	PR	117851.68	20.53

Insight:

Highest Total Freight Value: (SP) has the highest total freight value, amounting to approximately 718,723.07 units of currency. This is expected, given that SP is one of the most populous and economically significant states in the dataset.

Lowest Total Freight Value: (RR) has the lowest total freight value, with a value of approximately 2,235.19 units of currency. Roraima is one of the smaller states in terms of population and economic activity.

Average Freight Price Variations: There are notable variations in average freight prices across states. (PB) has the highest average freight price, approximately 42.72 units of currency, indicating that customers in this state, on average, pay a higher freight cost per order.

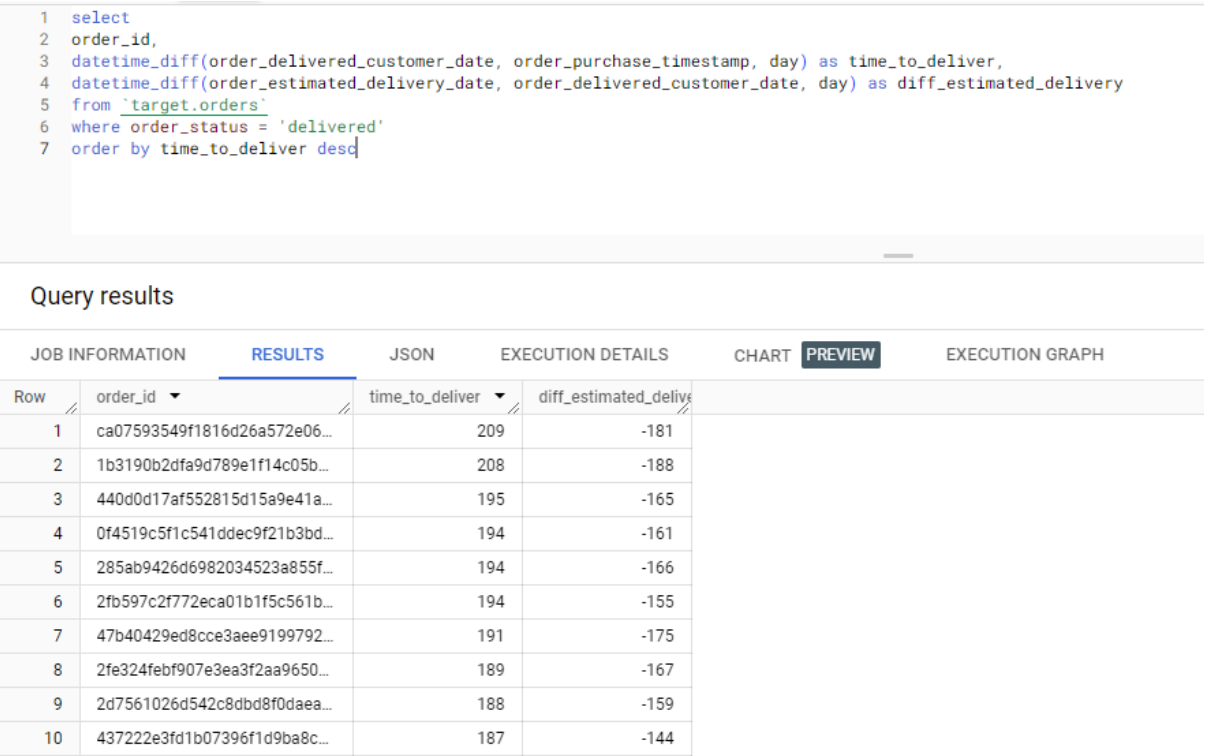
Lowest Average Freight Price: (SP), despite having the highest total freight value, has one of the lowest average freight prices, approximately 15.15 units of currency. This suggests that although there is a high volume of freight in SP, the average cost per order is relatively lower compared to other states.

Freight Cost Efficiency: States with lower average freight prices may indicate more efficient logistics networks or shorter shipping distances, resulting in lower shipping costs for customers.

Freight Price Impact: High average freight prices, such as in (PB) and (RR), may impact customer purchasing decisions, as customers in these states may be more sensitive to shipping costs when making online purchases.

V Analysis based on sales, freight and delivery time.

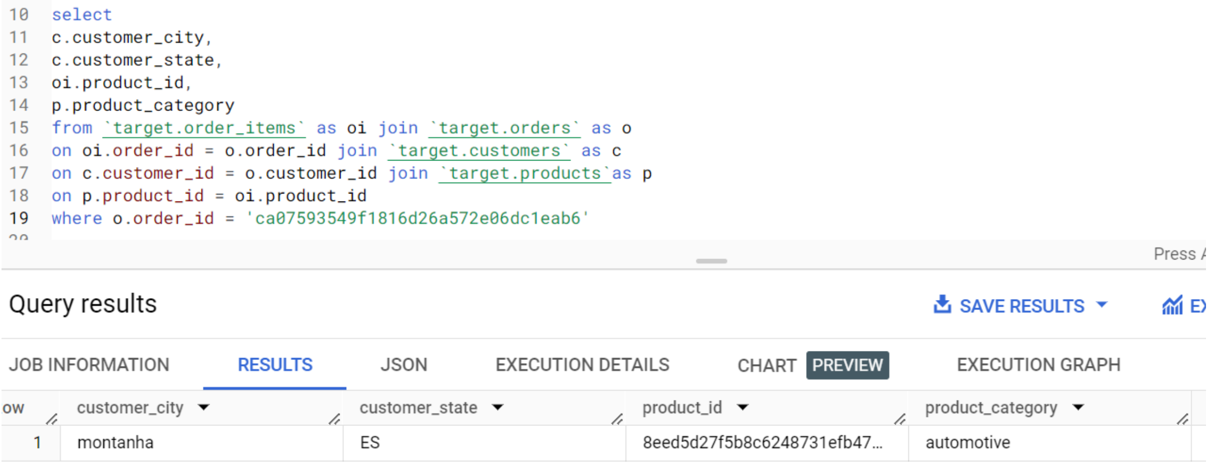
A. Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.



```sql
select
order_id,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
from `target.orders`
where order_status = 'delivered'
order by time_to_deliver desc
```

Query results

Row	order_id	time_to_deliver	diff_estimated_delive
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	0f4519c5f1c541ddec9f21b3bd...	194	-161
5	285ab9426d6982034523a855f...	194	-166
6	2fb597c2f772eca01b1f5c561b...	194	-155
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	437222e3fd1b07396f1d9ba8c...	187	-144



```sql
select
c.customer_city,
c.customer_state,
oi.product_id,
p.product_category
from `target.order_items` as oi join `target.orders` as o
on oi.order_id = o.order_id join `target.customers` as c
on c.customer_id = o.customer_id join `target.products`as p
on p.product_id = oi.product_id
where o.order_id = 'ca07593549f1816d26a572e06dc1eab6'
```

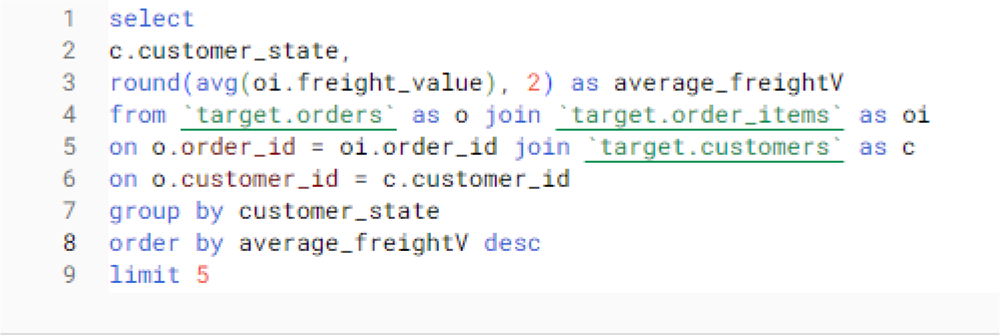
Query results

ow	customer_city	customer_state	product_id	product_category
1	montanha	ES	8eed5d27f5b8c6248731efb47...	automotive

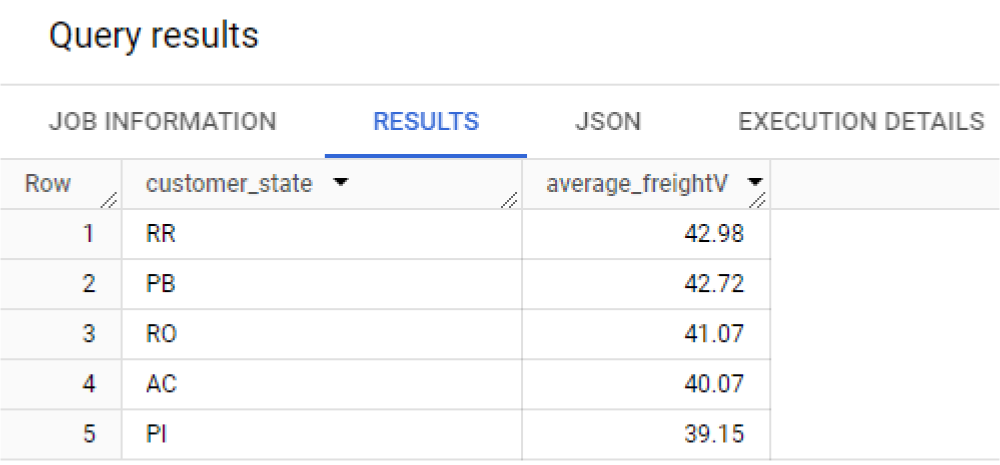
Insight:

The maximum order delivery time taken is 209 days and minimum is 1 day. The reason behind the long days of delivery is not clear. But the order is from Montanha city , ES state having product category from automotive.

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



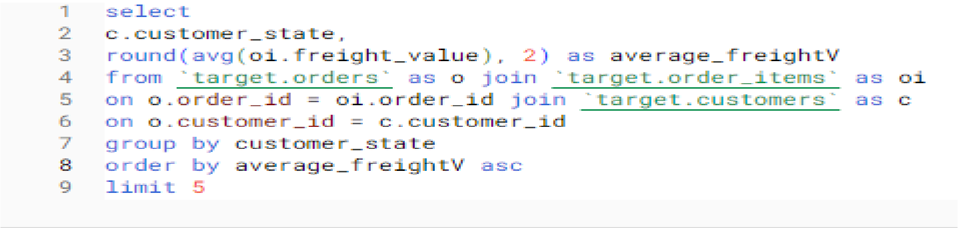
```sql
select
c.customer_state,
round(avg(oi.freight_value), 2) as average_freightV
from `target.orders` as o join `target.order_items` as oi
on o.order_id = oi.order_id join `target.customers` as c
on o.customer_id = c.customer_id
group by customer_state
order by average_freightV desc
limit 5
```



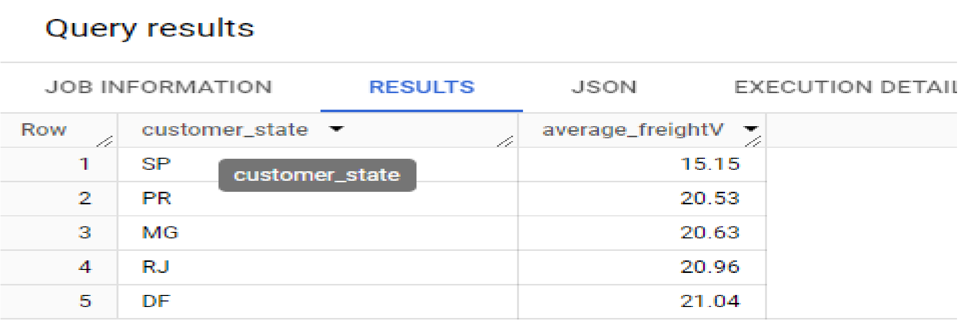
Query results

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

b. lowest values



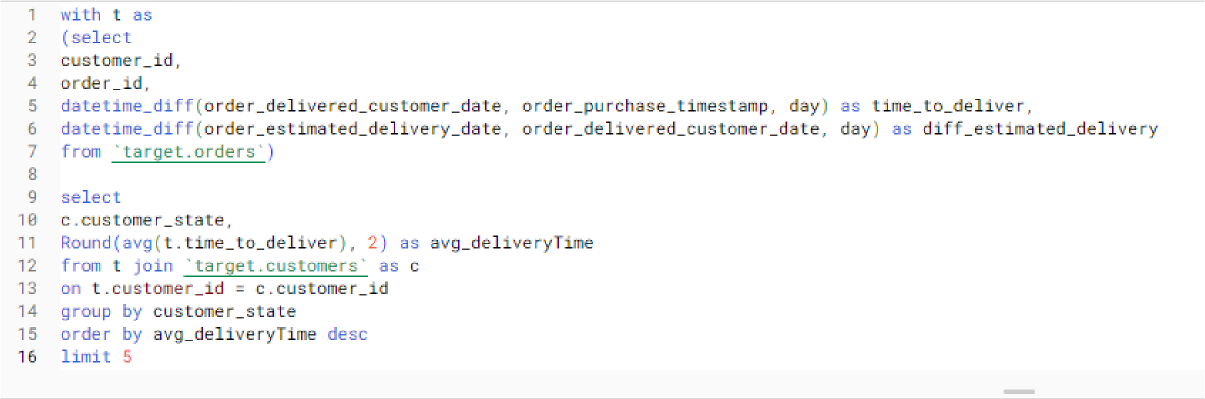
```sql
select
c.customer_state,
round(avg(oi.freight_value), 2) as average_freightV
from `target.orders` as o join `target.order_items` as oi
on o.order_id = oi.order_id join `target.customers` as c
on o.customer_id = c.customer_id
group by customer_state
order by average_freightV asc
limit 5
```



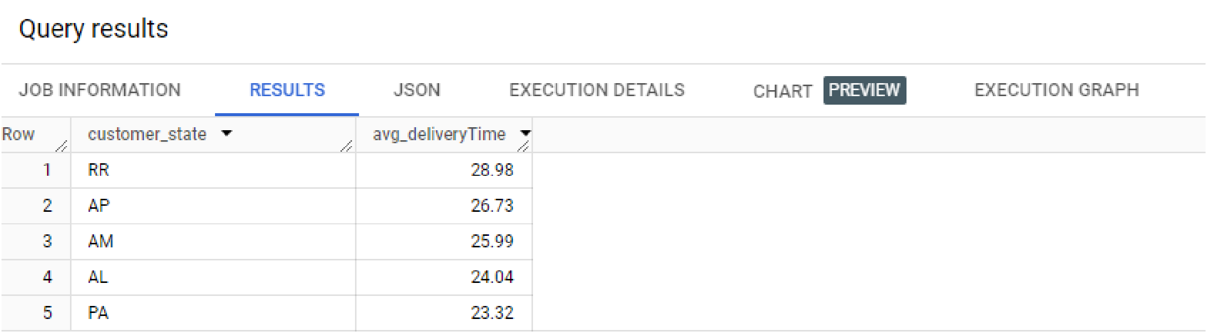
Query results

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

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



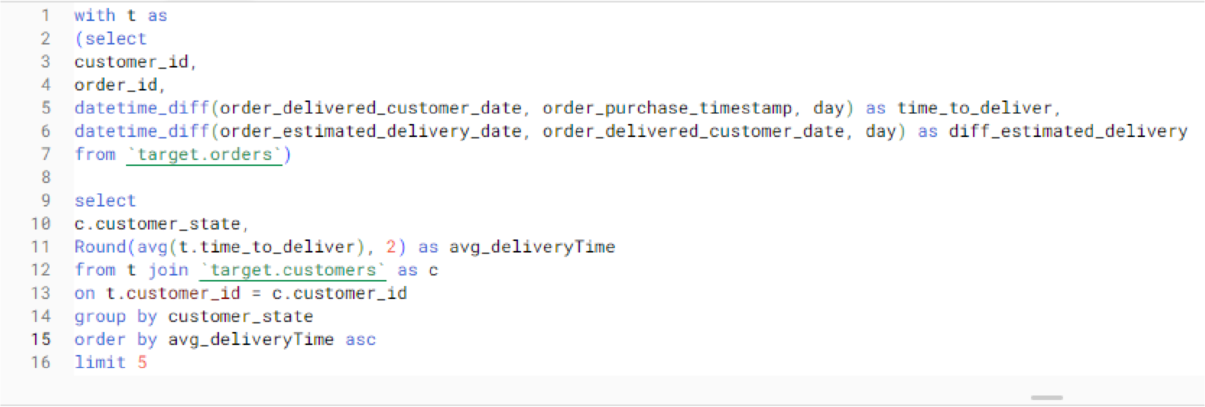
```sql
with t as
(select
customer_id,
order_id,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
from `target.orders`)

select
c.customer_state,
Round(avg(t.time_to_deliver), 2) as avg_deliveryTime
from t join `target.customers` as c
on t.customer_id = c.customer_id
group by customer_state
order by avg_deliveryTime desc
limit 5
```



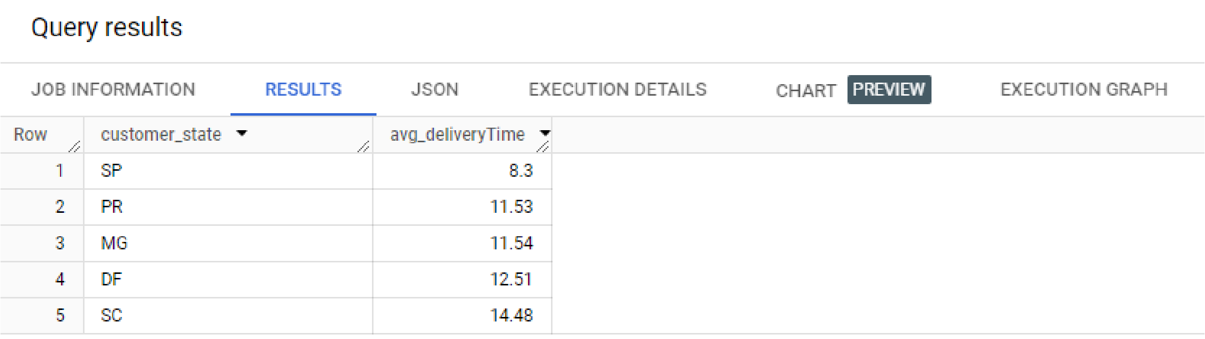
Query results

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

c lowest



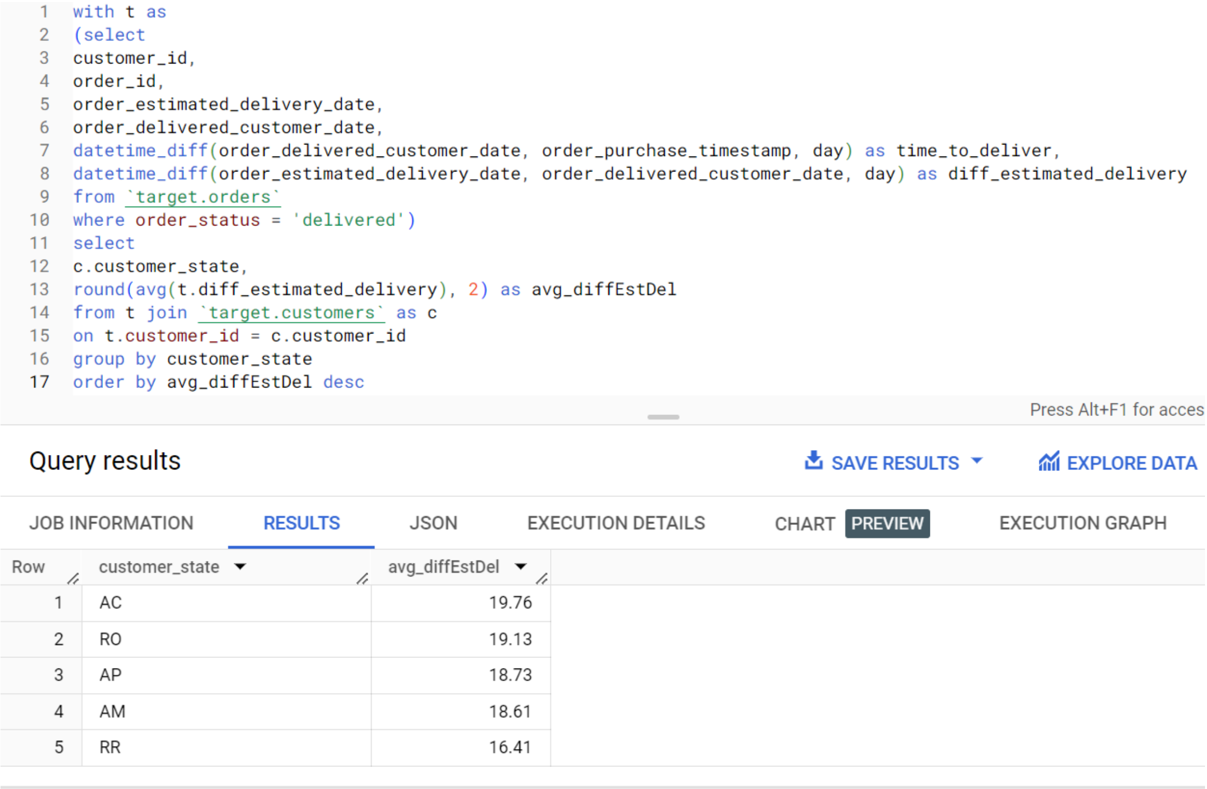
```sql
with t as
(select
customer_id,
order_id,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
from `target.orders`)

select
c.customer_state,
Round(avg(t.time_to_deliver), 2) as avg_deliveryTime
from t join `target.customers` as c
on t.customer_id = c.customer_id
group by customer_state
order by avg_deliveryTime asc
limit 5
```



Query results

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

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

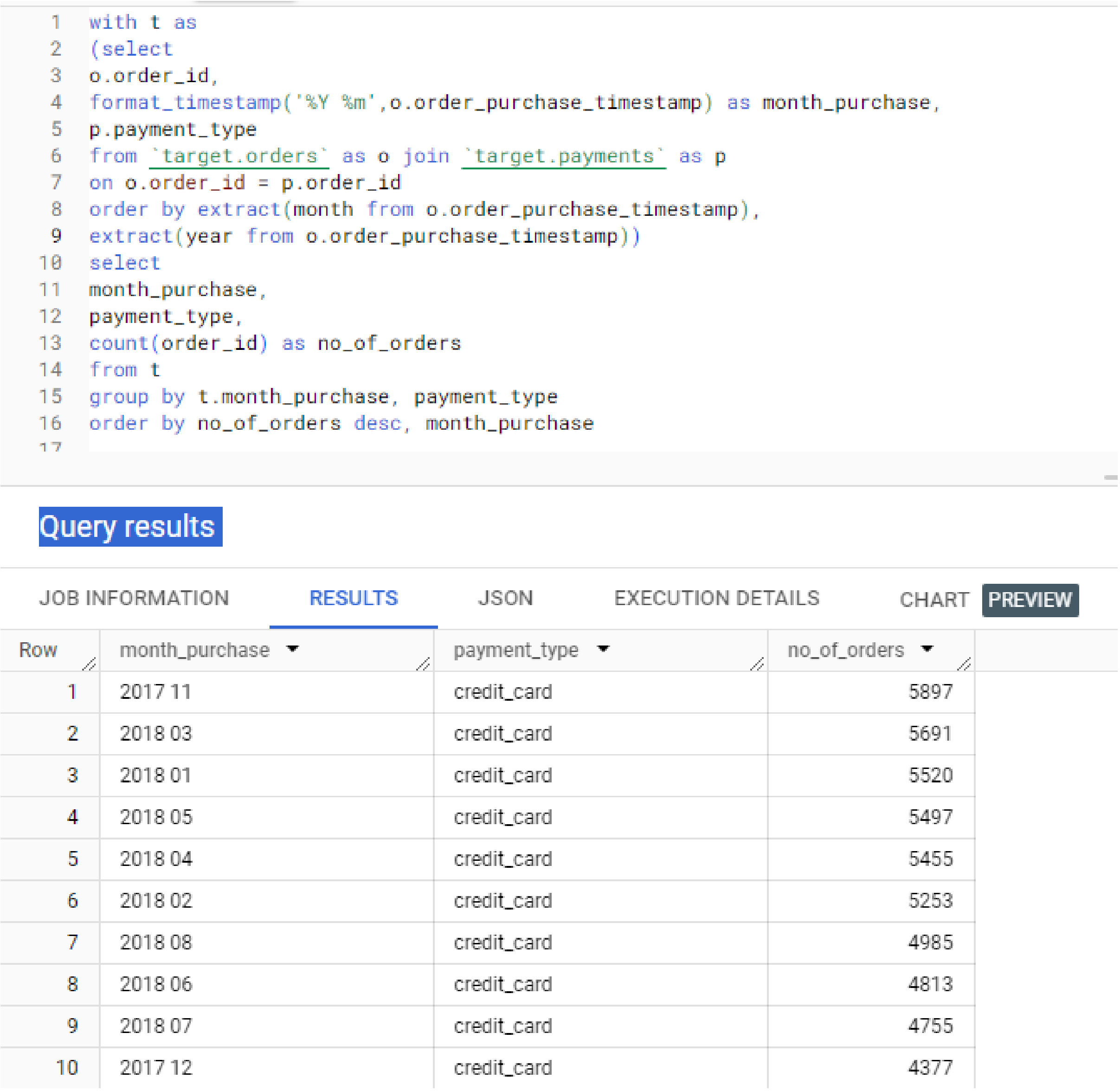


```sql
with t as
(select
customer_id,
order_id,
order_estimated_delivery_date,
order_delivered_customer_date,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
from `target.orders`
where order_status = 'delivered')
select
c.customer_state,
round(avg(t.diff_estimated_delivery), 2) as avg_diffEstDel
from t join `target.customers` as c
on t.customer_id = c.customer_id
group by customer_state
order by avg_diffEstDel desc
```

Row	customer_state	avg_diffEstDel
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

VI. **Analysis based on the payments:**

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



```
with t as
(select
o.order_id,
format_timestamp('%Y %m',o.order_purchase_timestamp) as month_purchase,
p.payment_type
from `target.orders` as o join `target.payments` as p
on o.order_id = p.order_id
order by extract(month from o.order_purchase_timestamp),
extract(year from o.order_purchase_timestamp))
select
month_purchase,
payment_type,
count(order_id) as no_of_orders
from t
group by t.month_purchase, payment_type
order by no_of_orders desc, month_purchase
```

Row	month_purchase	payment_type	no_of_orders
1	2017 11	credit_card	5897
2	2018 03	credit_card	5691
3	2018 01	credit_card	5520
4	2018 05	credit_card	5497
5	2018 04	credit_card	5455
6	2018 02	credit_card	5253
7	2018 08	credit_card	4985
8	2018 06	credit_card	4813
9	2018 07	credit_card	4755
10	2017 12	credit_card	4377

Insight:

Monthly Trends:

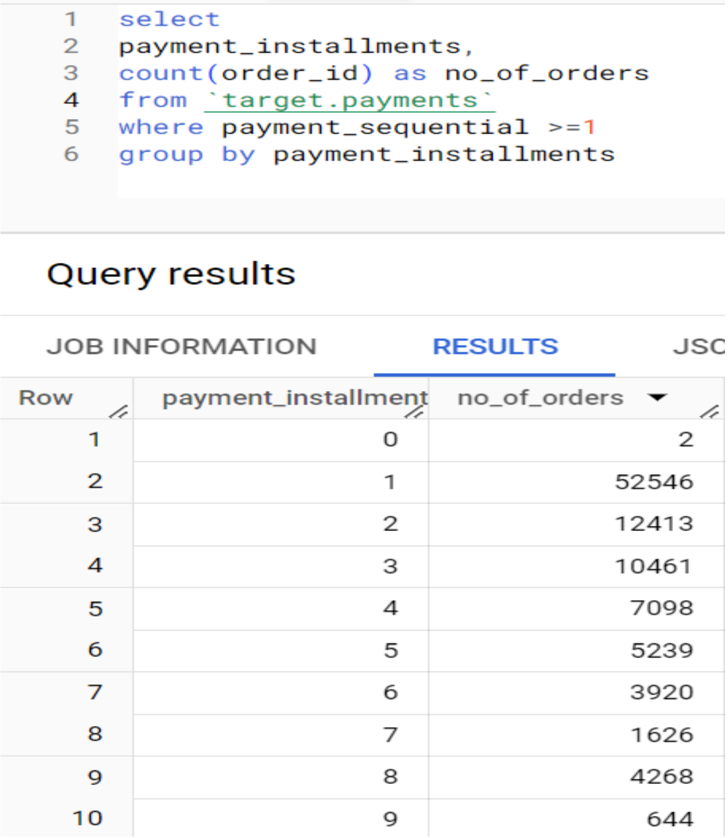
November 2017 had the highest overall number of orders, primarily driven by credit card orders.

Orders in January and February 2017 were primarily credit card and UPI orders.

July 2018 saw increased activity in both credit card and UPI orders, while voucher orders also had a slight increase.

Debit card orders saw a notable increase in July 2018, reaching their highest point.

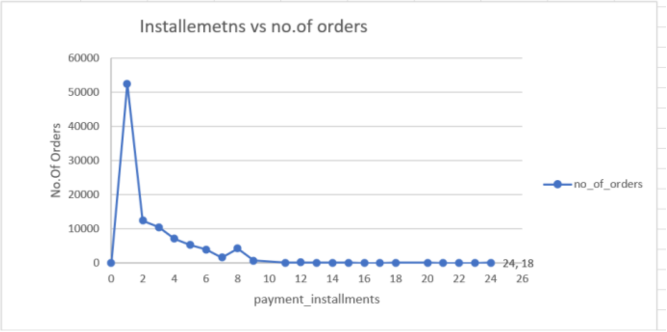
B.Find the no. of orders placed on the basis of the payment installments that have been paid.



```
select
payment_installments,
count(order_id) as no_of_orders
from `target.payments`
where payment_sequential >=1
group by payment_installments
```

Query results

Row	payment_installment	no_of_orders
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



Insight:

Installment vs. Orders Distribution:

Most orders (52,546) were placed without any installments (0 installments).

The number of orders decreased as the number of installments increased.

There was a significant drop in the number of orders with more than 1 installment