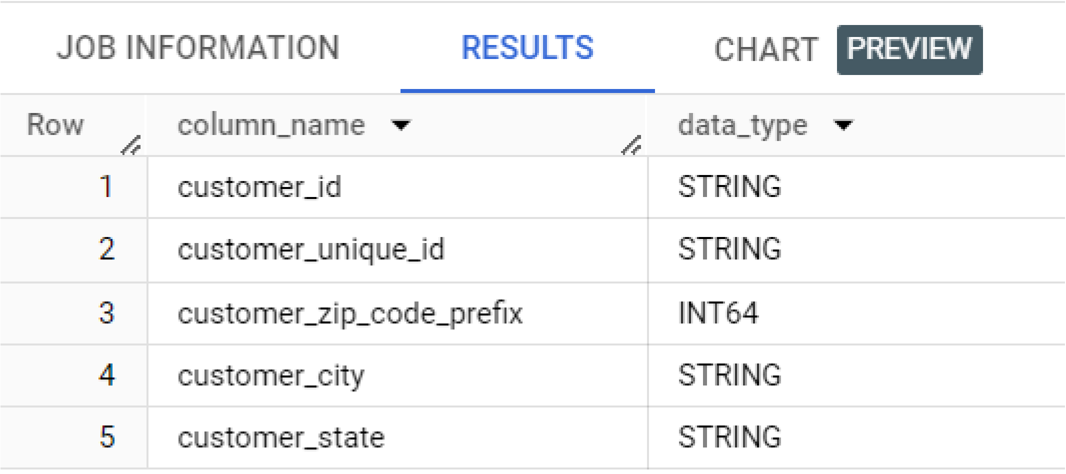
Target SQL Business Case

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.

Ans.

```
SELECT
column_name,
data_type
FROM scaler-dsml-sql-399503.target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name='customers';
```



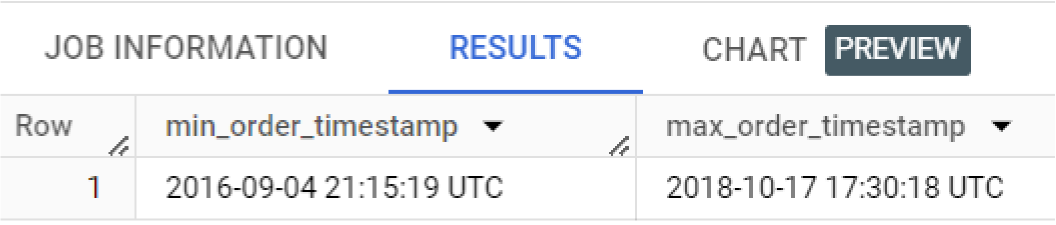
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Insights: Customer Table holds more of string datatype compared to others.

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

Ans.

```
SELECT
  MIN(order_purchase_timestamp) AS min_order_timestamp,
  MAX(order_purchase_timestamp) AS max_order_timestamp
FROM
  target.orders;
```



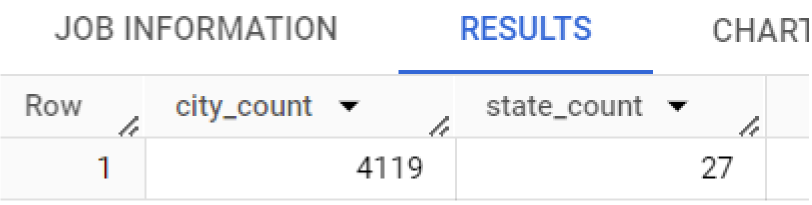
Row	min_order_timestamp	max_order_timestamp
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights: The orders in the dataset were placed within a time range from September 2016 to October 2018. This time range gives us an understanding of the duration over which the orders were made.

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

Ans.

```
SELECT
COUNT(DISTINCT c.customer_city ) as city_count,
COUNT(DISTINCT c.customer_state) as state_count
FROM target.customers c
INNER JOIN target.orders o on o.customer_id=c.customer_id
```



Row	city_count	state_count
1	4119	27

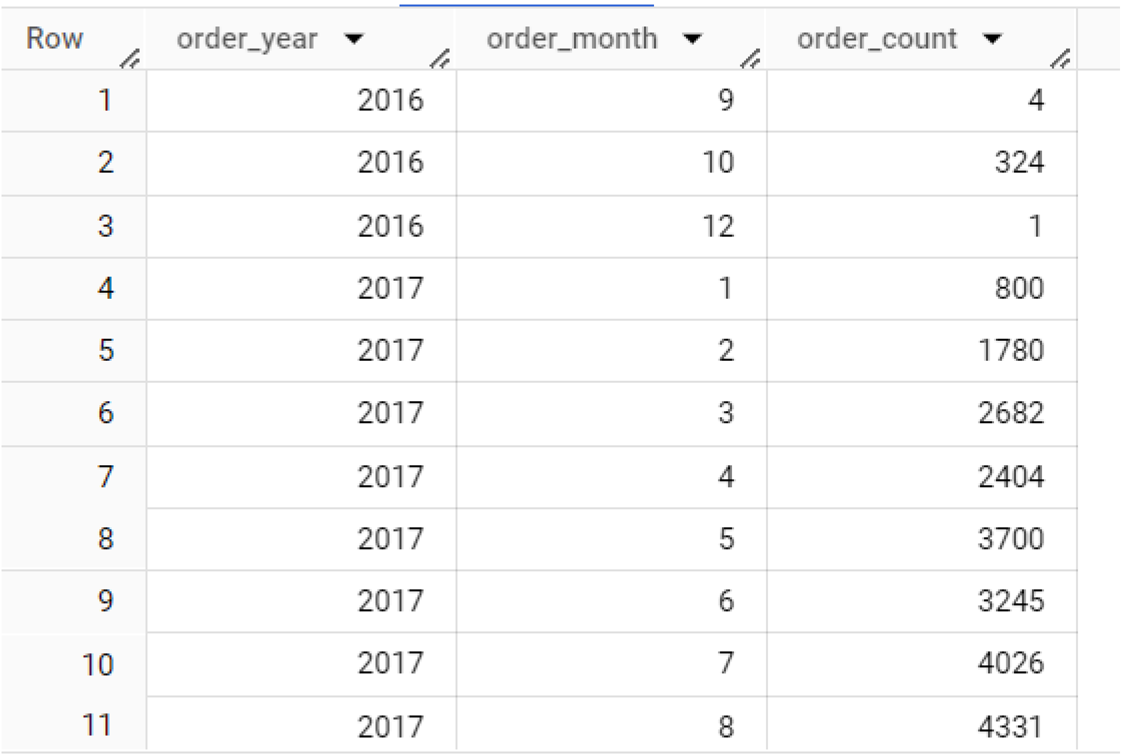
Insights: The query results show the count of unique cities and states from which customers placed orders during the specified time period. This information can be valuable for understanding the geographical distribution of customers and their locations at a specific time frame.

II. In-depth Exploration:

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

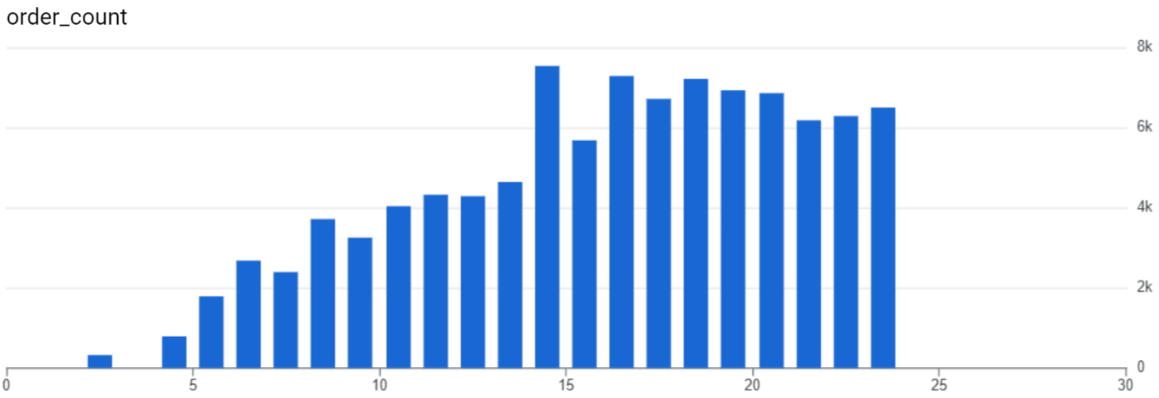
Ans:

```
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
  COUNT(DISTINCT order_id) AS order_count
FROM
  target.orders
GROUP BY
  order_year, order_month
ORDER BY
  order_year, order_month;
```



Row	order_year	order_month	order_count
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
11	2017	8	4331

Chart:



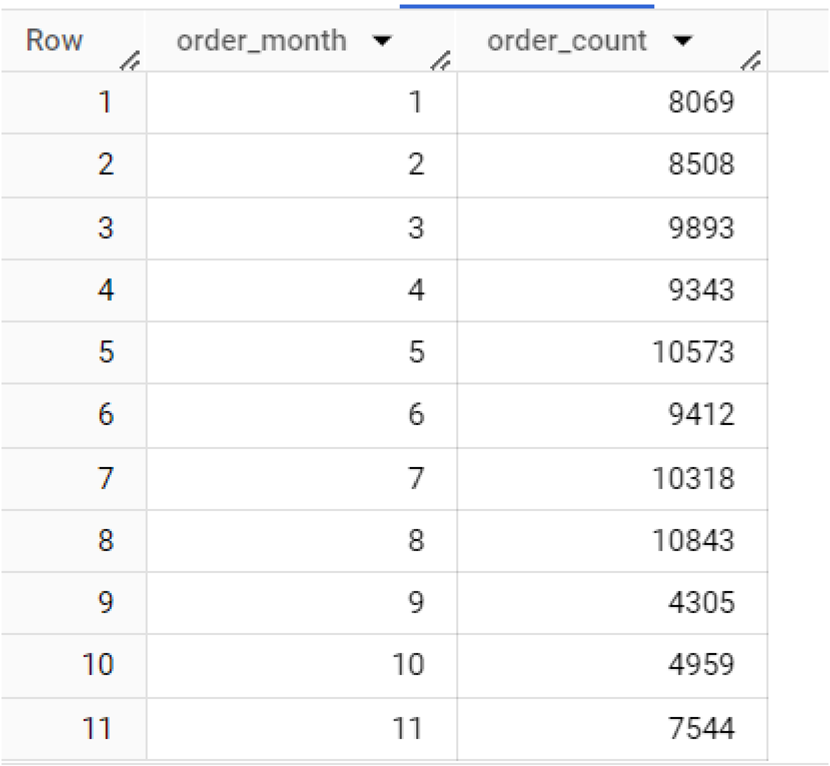
Insights:

1. The number of orders placed has shown a consistent upward trend over the past few years. The order counts have generally increased from 2016 to 2018.

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

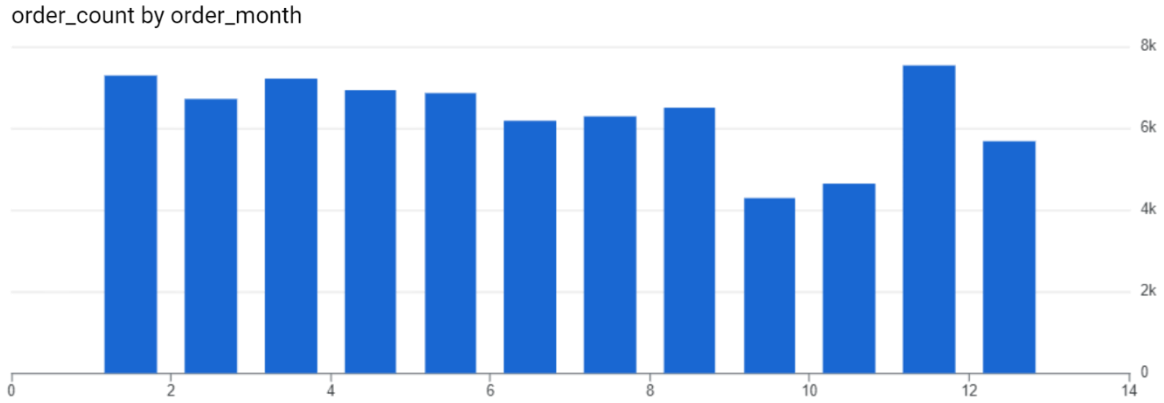
Ans:

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
  COUNT(DISTINCT order_id) AS avg_order_count
FROM
  target.orders
GROUP BY
  order_month
ORDER BY
  order_month;
```



Row	order_month	order_count
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

Chart:



Insights:

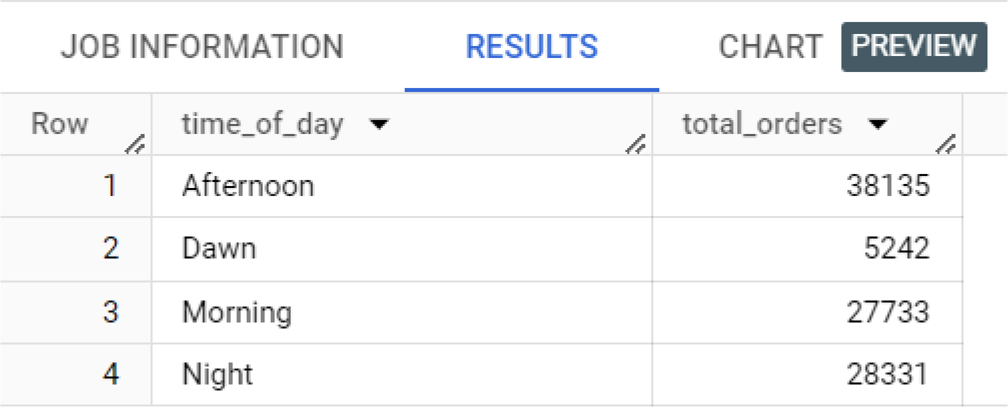
There are fluctuations in order counts across months, with certain months experiencing higher order volumes than others. For example, there is a significant spike in orders during November and December of 2017, which is likely due to holiday season shopping

C. 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**

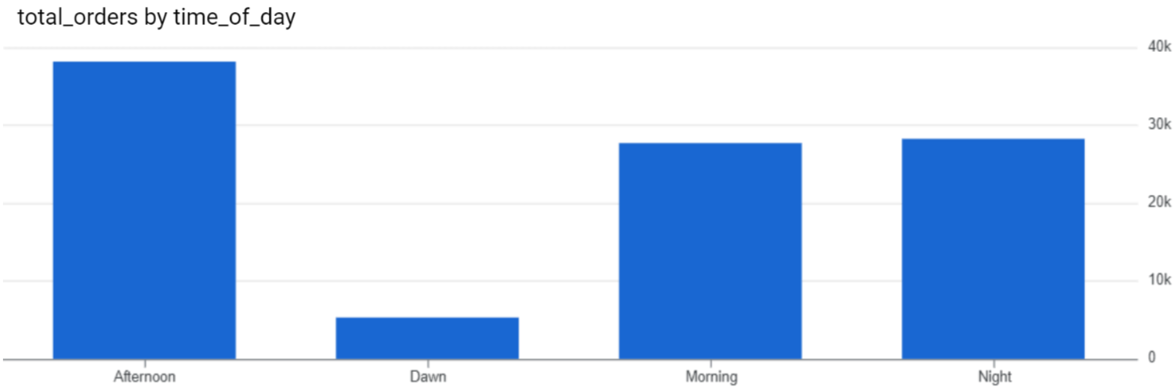
Ans:

```
WITH order_times AS
(
  SELECT
    EXTRACT(HOUR FROM order_purchase_timestamp) AS order_hour,
    COUNT(DISTINCT order_id) AS order_count
  FROM
    target.orders
  GROUP BY
    order_hour
  ORDER BY
    order_hour
)
SELECT
  CASE
    WHEN order_hour >= 0 AND order_hour <= 6 THEN 'Dawn'
    WHEN order_hour >= 7 AND order_hour <= 12 THEN 'Morning'
    WHEN order_hour >= 13 AND order_hour <= 18 THEN 'Afternoon'
    WHEN order_hour >= 19 AND order_hour <= 23 THEN 'Night'
    END AS time_of_day,
  SUM(order_count) AS total_orders
FROM
  order_times
GROUP BY
  time_of_day
ORDER BY
  time_of_day;
```



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

Chart:



<u>Insights:</u>

1. The "Afternoon" category has the highest total number of orders, with 38,135 orders. This suggests that a significant portion of customers in Brazil tends to place their orders in the afternoon, indicating a peak in shopping activity during this time.

2. The "Morning" and "Night" categories also show substantial order counts, with 27,733 orders in the morning and 28,331 orders at night. This implies that there is consistent shopping activity in the morning and late at night.

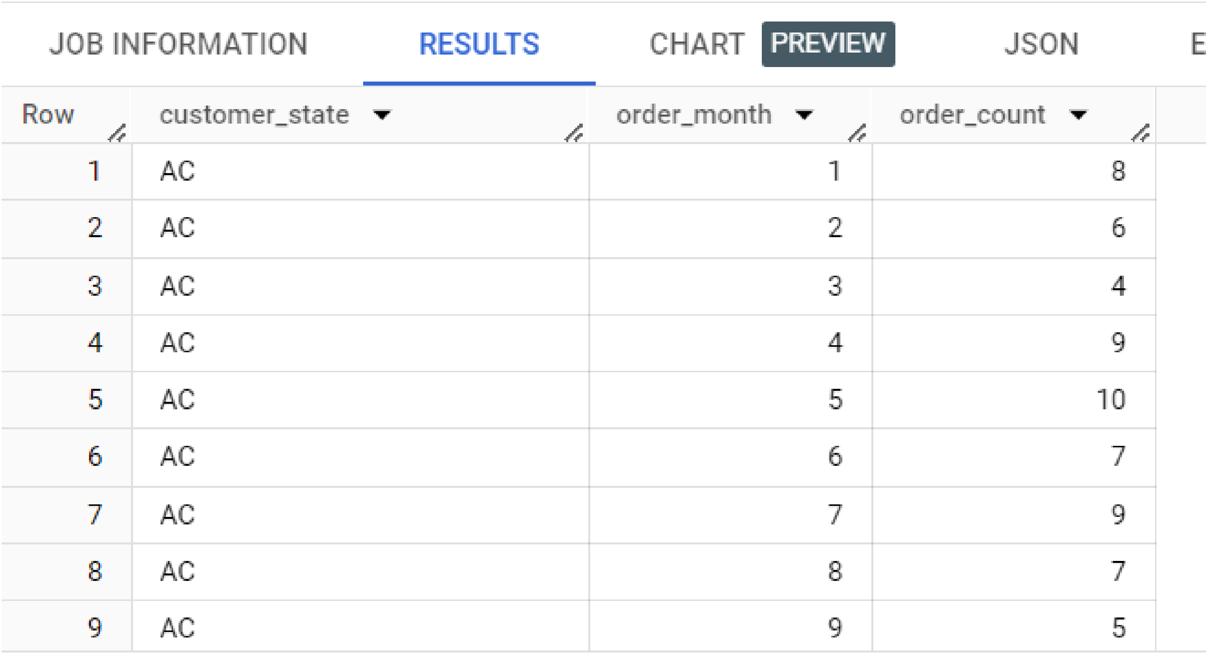
3. The "Dawn" category has the lowest order count with 5,242 orders. While it has fewer orders compared to other times, it's not negligible, indicating that some customers in Brazil do place orders during the early hours.

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

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

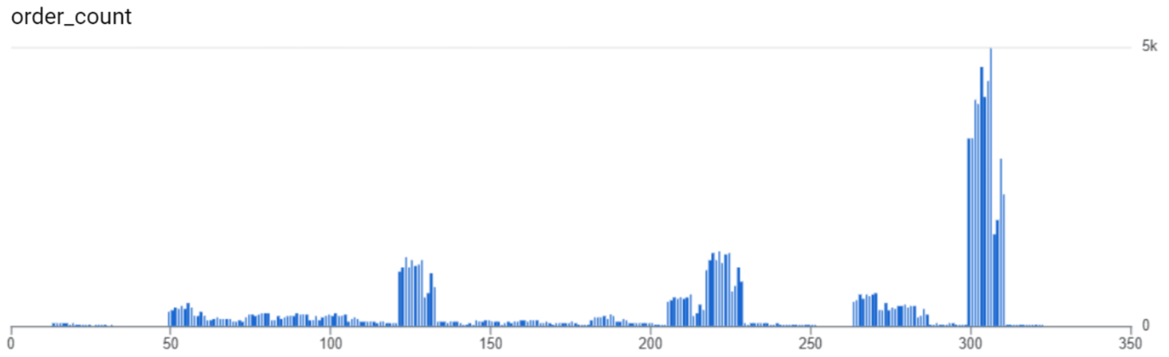
Ans:

```
WITH monthly_order_counts AS (
  SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
    customer_state,
    COUNT(DISTINCT order_id) AS order_count
  FROM
   target.orders o join target.customers c using(customer_id)
   GROUP BY
    customer_state,order_month
  ORDER BY
    customer_state,order_month
)
SELECT
  customer_state,
  order_month,
  order_count
FROM
  monthly_order_counts
ORDER BY
  customer_state,order_month;
```



Row	customer_state	order_month	order_count
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

Chart:



Insights:

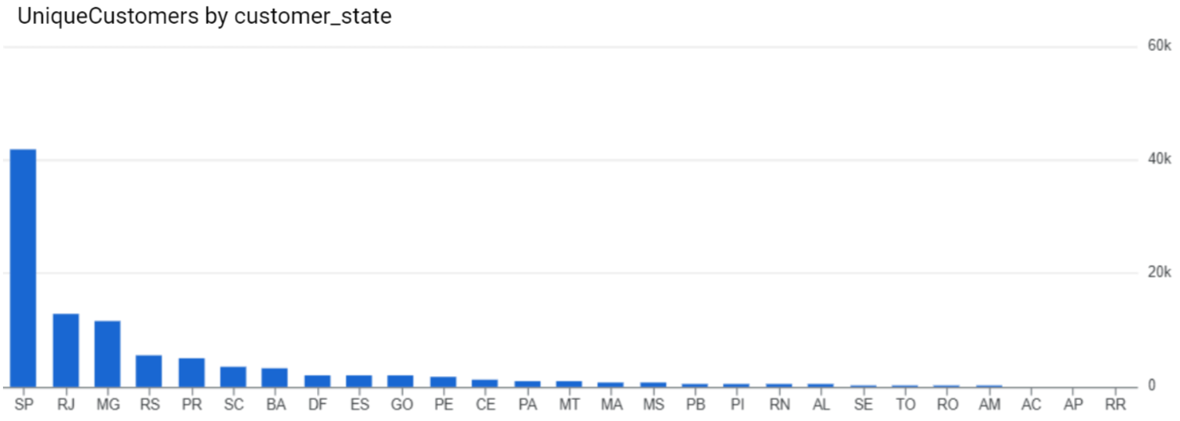
1. By analyzing the order counts, you can gain insights into the regional trends in customer orders. For example, states like SP, MG, and RJ tend to have consistently high order counts throughout the year.
2. There is a sudden drop for state like AM in orders during the last few months of the year .

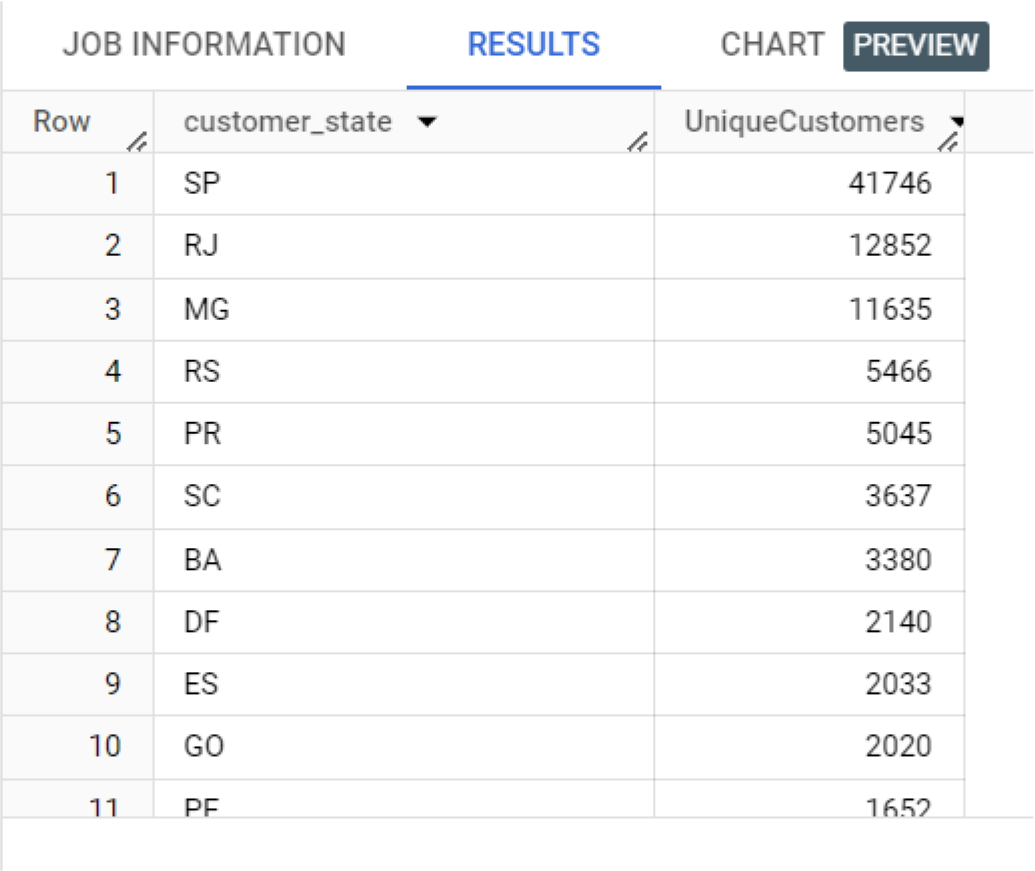
B. How are the customers distributed across all the states?

Ans:

```
SELECT customer_state, COUNT(DISTINCT customer_id) AS UniqueCustomers
FROM target.customers
GROUP BY customer_state
ORDER BY UniqueCustomers DESC;
```

Chart:





Row	customer_state	UniqueCustomers
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
11	PE	1652

Insights:

1. The states of SP, RJ, and MG have a significantly higher number of unique customers compared to other states.
2. States with lower populations like AC, RR and AP have the fewest unique customers.
3. States like BA, DF and ES have a moderate number of unique customers.

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

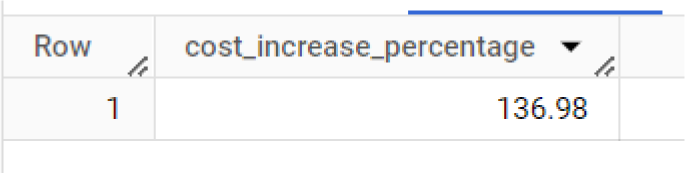
A. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

ANS:

```
WITH OrderCosts AS (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    sum(payment_value) AS total_order_cost
  FROM
    target.orders o join target.payments p using(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)

)

SELECT
  round(((oc2.total_order_cost - oc1.total_order_cost) / oc1.total_order_cost) *
100,2) AS cost_increase_percentage
FROM
  OrderCosts oc1
  cross join
  OrderCosts oc2
where
  oc1.order_year = 2017
  AND oc2.order_year = 2018
```



Row	cost_increase_percentage
1	136.98

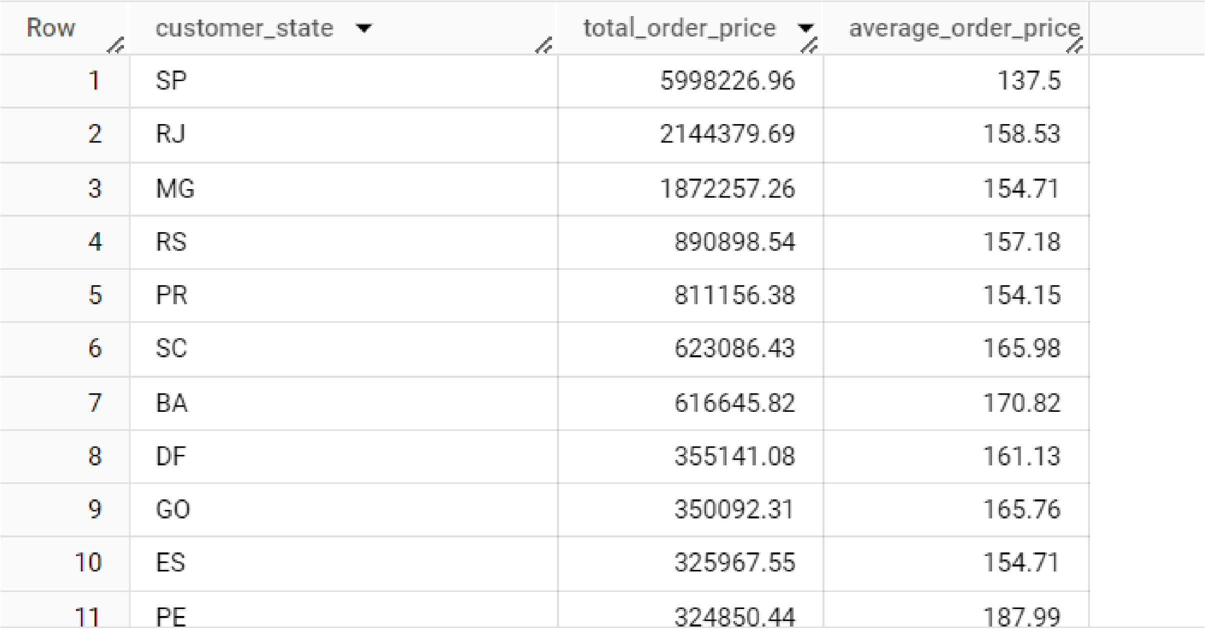
Insights:

The percentage increase in the cost of orders from 2017 to 2018 for the months between January to August is approximately 136.98%. The cost of orders in the first eight months of 2018 increased significantly compared to the same period in 2017.

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

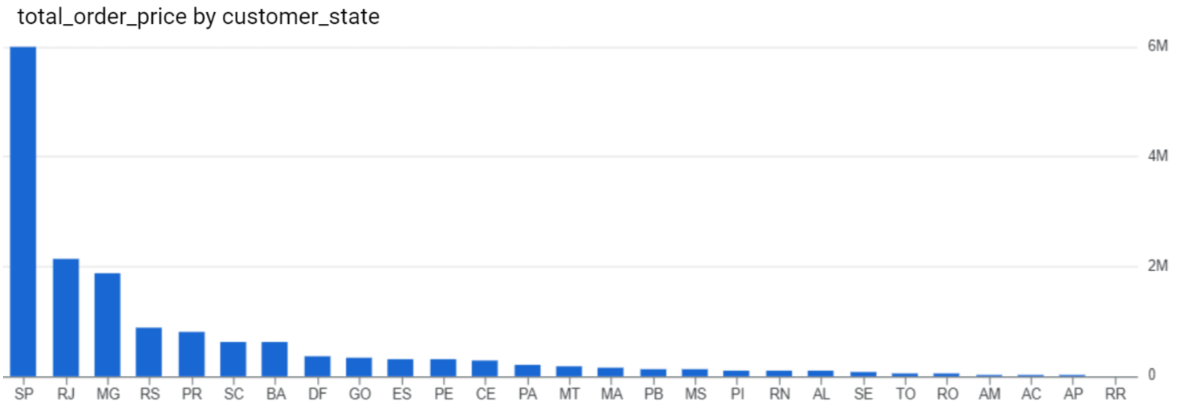
Ans:

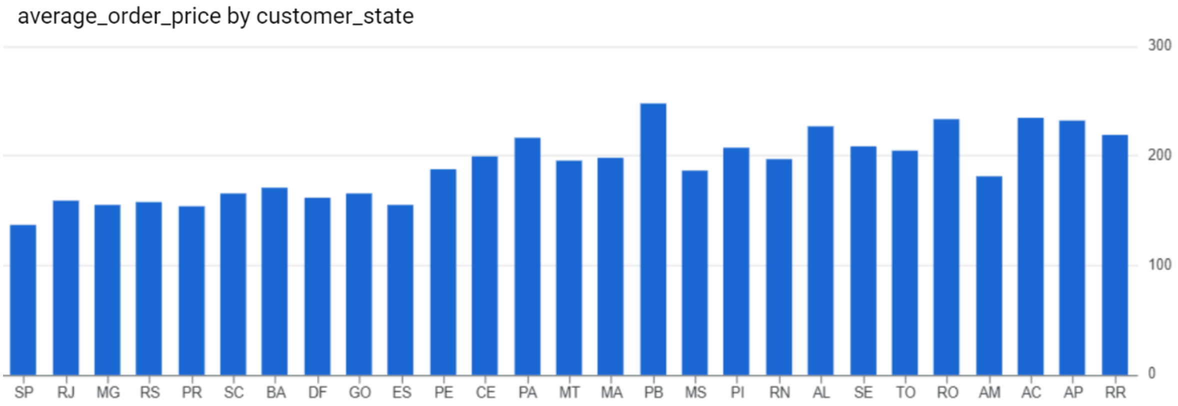
```
SELECT
    c.customer_state,
    round(SUM(p.payment_value),2) AS total_order_price,
    round(AVG(p.payment_value),2) AS average_order_price
FROM
    target.orders o
JOIN
    target.customers c using(customer_id)
JOIN
    target.payments p using(order_id)
GROUP BY
    c.customer_state
ORDER BY
    2 desc;
```



Row	customer_state	total_order_price	average_order_price
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76
10	ES	325967.55	154.71
11	PE	324850.44	187.99

Charts:





Insights:

1. SP has the highest total order price, which is approximately 5,998,226.96. This indicates that the highest sales revenue is generated from SP.

2. PB has the highest average order price, approximately 248.33. This suggests that customers in PB, on average, spend more per order compared to other states.

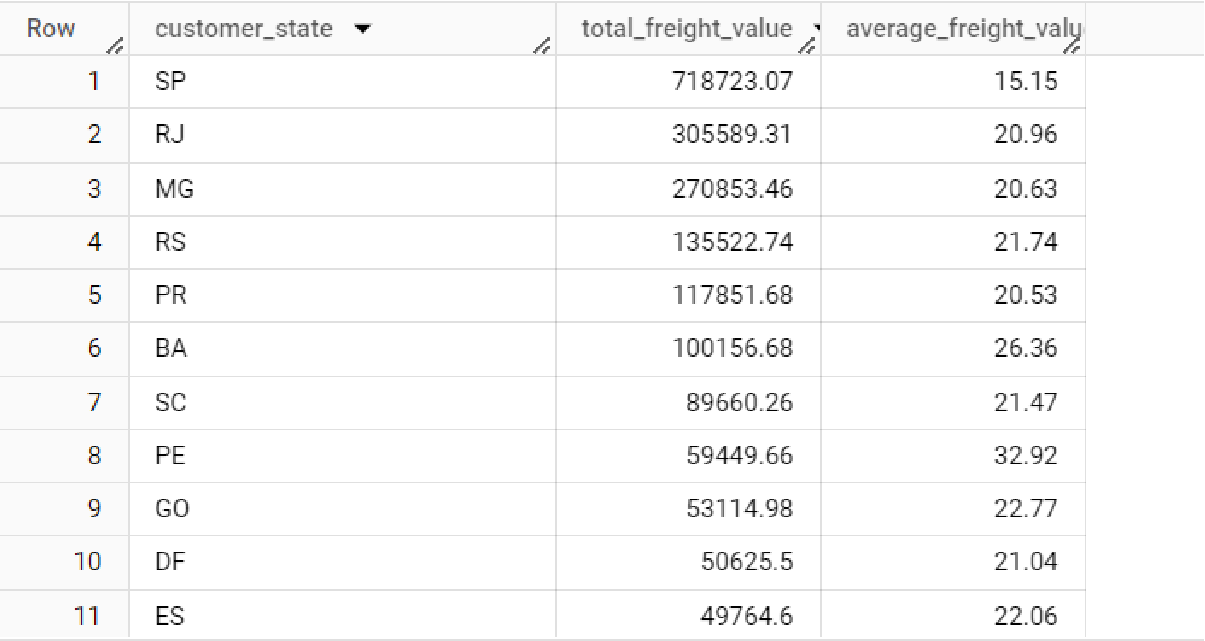
3. The average order prices vary significantly across different states.

4. Highest total order price and highest average price are not having for same state.

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

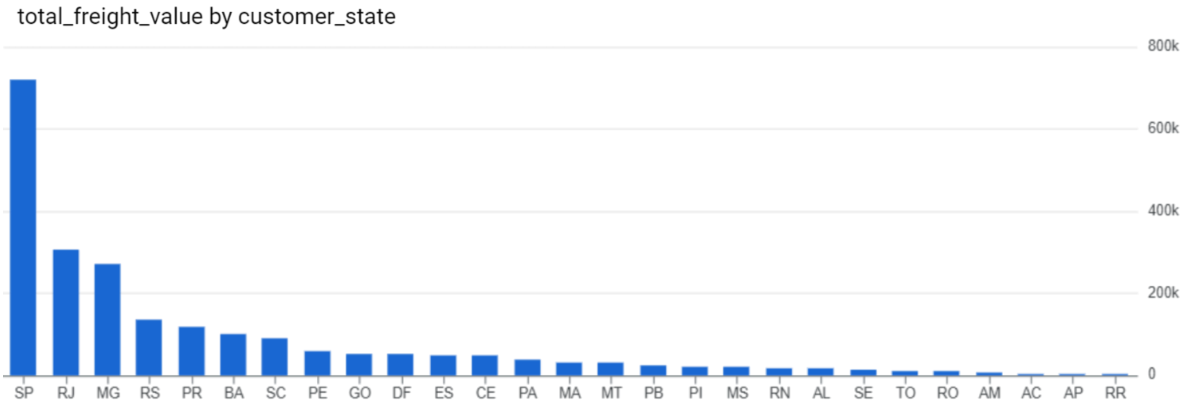
Ans:

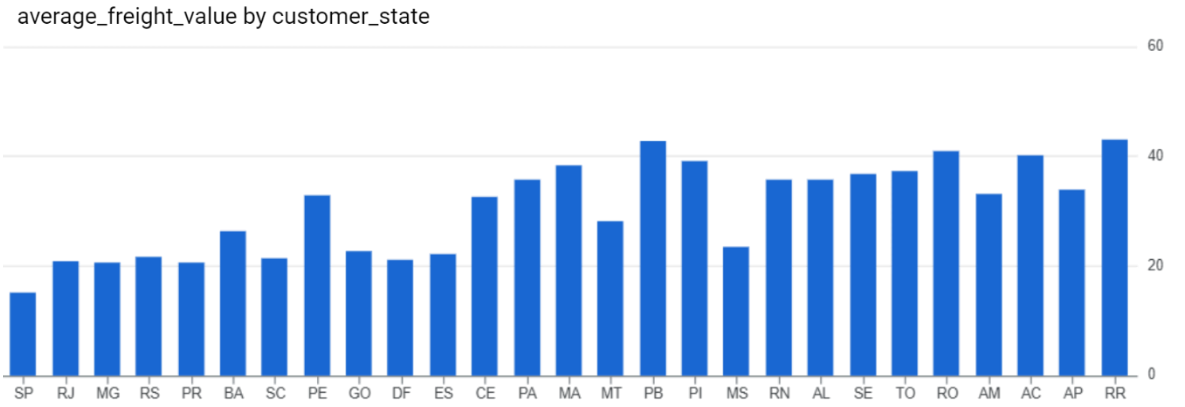
```
SELECT
    c.customer_state AS customer_state,
    round(SUM(oi.freight_value),2) AS total_freight_value,
    round(AVG(oi.freight_value),2) AS average_freight_value
FROM
    target.orders o
JOIN
    target.order_items oi ON o.order_id = oi.order_id
JOIN
    target.customers c ON o.customer_id = c.customer_id
GROUP BY
    c.customer_state
ORDER BY
    2 desc;
```



Row	customer_state	total_freight_value	average_freight_valu
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04
11	ES	49764.6	22.06

Charts:





Insights:

1. The total freight cost varies significantly across different states. SP has the highest total freight cost. However states like RJ, MG and RS also have considerable total freight costs.

2. We can see that states with lowest total freight costs has the highest average freight costs.

3. PB has the highest average freight cost.

4. States with higher average freight costs may require new shipping options.

5. Higher average freight costs could also indicate a willingness of customers to pay more for shipping.

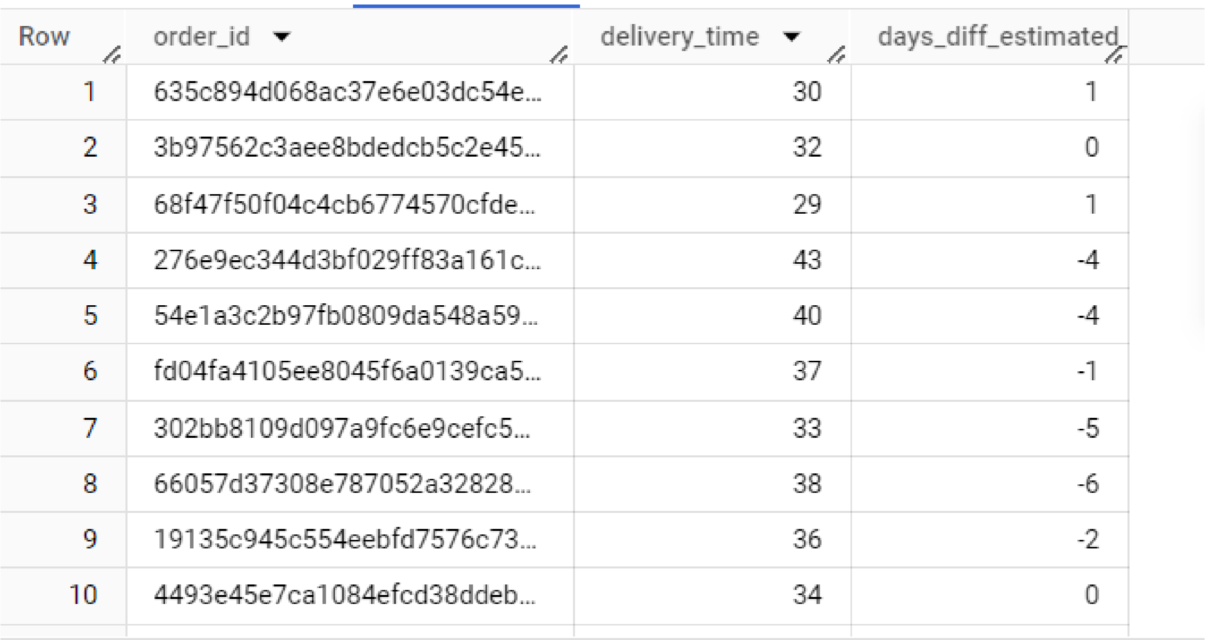
6. The freight cost directly affects the profit margins of e-commerce businesses.

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.

Ans:

```
SELECT
    order_id,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS
delivery_time,
    DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,DAY) AS
days_diff_estimated_actual
FROM
    target.orders
WHERE
  order_status='delivered' ;
```



Row	order_id	delivery_time	days_diff_estimated
1	635c894d068ac37e6e03dc54e...	30	1
2	3b97562c3aee8bdedcb5c2e45...	32	0
3	68f47f50f04c4cb6774570cfde...	29	1
4	276e9ec344d3bf029ff83a161c...	43	-4
5	54e1a3c2b97fb0809da548a59...	40	-4
6	fd04fa4105ee8045f6a0139ca5...	37	-1
7	302bb8109d097a9fc6e9cefc5...	33	-5
8	66057d37308e787052a32828...	38	-6
9	19135c945c554eebfd7576c73...	36	-2
10	4493e45e7ca1084efcd38ddeb...	34	0

Insights:

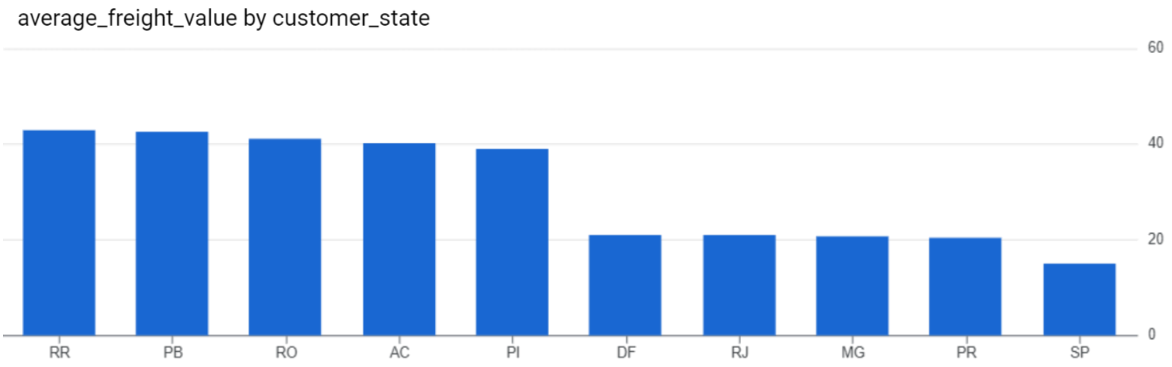
1. By observing delivery time column, we can say that for every order it takes atleast 30 days to deliver the order.

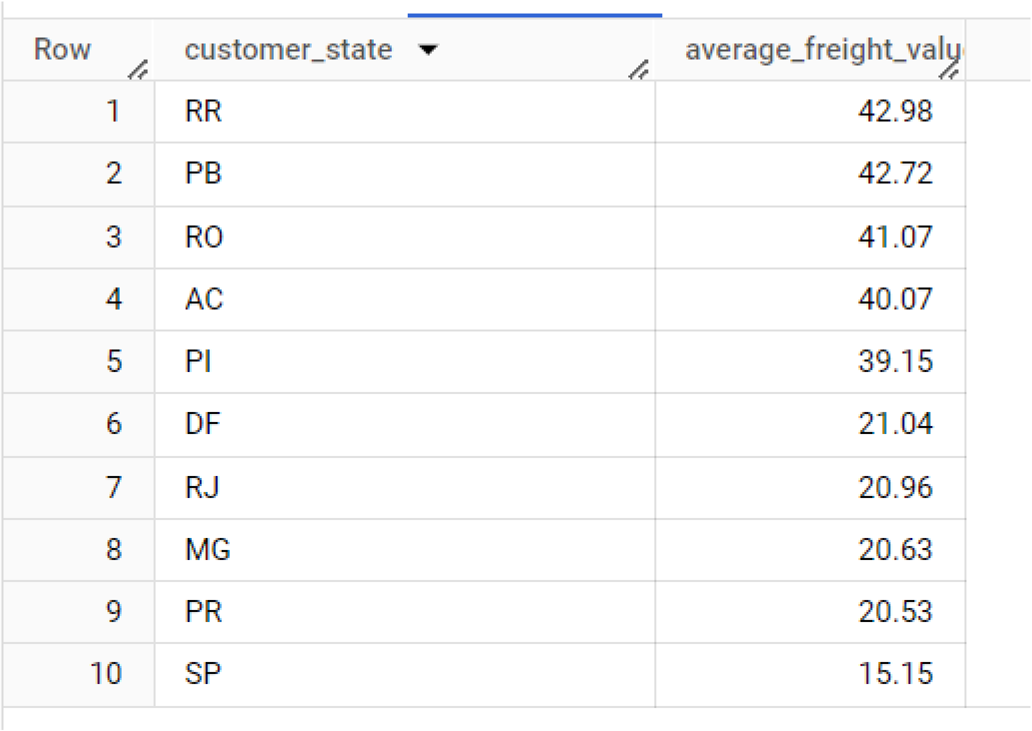
2. From days diff estimated column we can see that more are negative values we can say that order is delivered after the estimated delivery date.

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

Ans:

```
WITH TOP AS
(
SELECT
    c.customer_state ,
    round(AVG(oi.freight_value),2) AS average_freight_value
FROM
    target.orders o
JOIN
    target.order_items oi ON o.order_id = oi.order_id
JOIN
    target.customers c ON o.customer_id = c.customer_id
GROUP BY
    c.customer_state
ORDER BY
    2 desc
LIMIT 5
),
BOTTOM AS
(
SELECT
    c.customer_state ,
    round(AVG(oi.freight_value),2) AS average_freight_value
FROM
    target.orders o
JOIN
    target.order_items oi ON o.order_id = oi.order_id
JOIN
    target.customers c ON o.customer_id = c.customer_id
GROUP BY
    c.customer_state
ORDER BY
    2 asc
LIMIT 5
)
SELECT *
FROM TOP
UNION DISTINCT
SELECT *
FROM BOTTOM
ORDER BY average_freight_value desc
```





Row	customer_state	average_freight_valu
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15
6	DF	21.04
7	RJ	20.96
8	MG	20.63
9	PR	20.53
10	SP	15.15

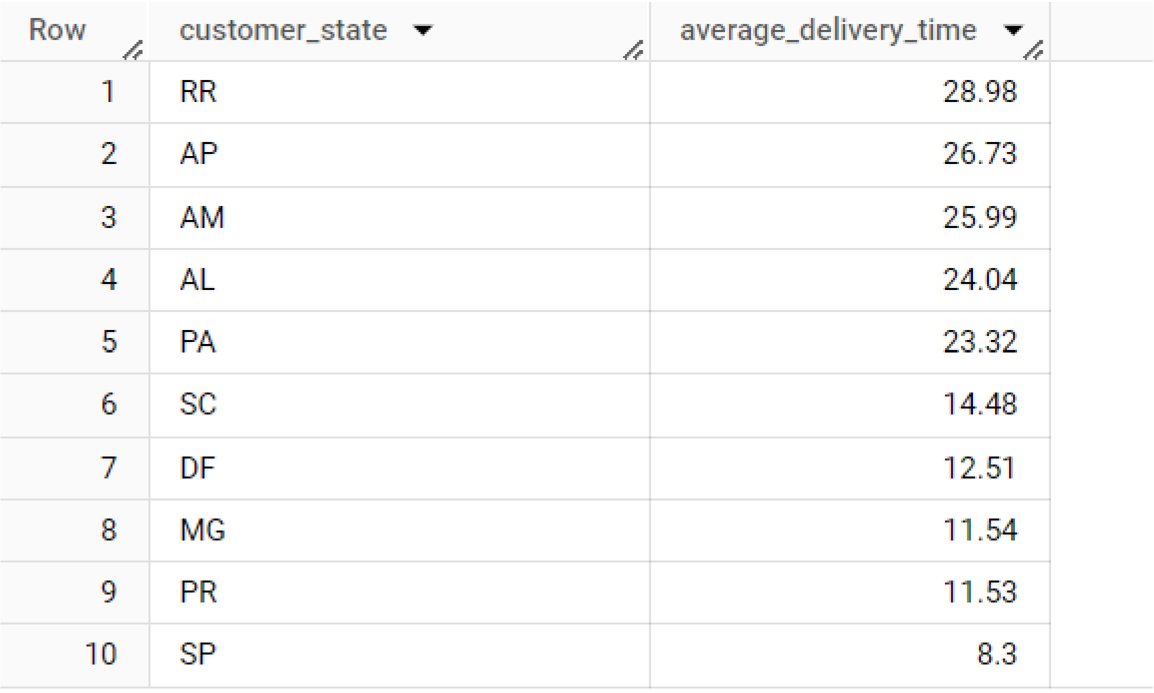
Insights:

States like RR, PB, and RO have the highest average freight values, indicating relatively higher shipping costs for customers. Conversely SP has the lowest average freight value, with MG and PR also having relatively lower shipping expenses. These insights can inform pricing and logistics decisions for e-commerce in these regions.

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

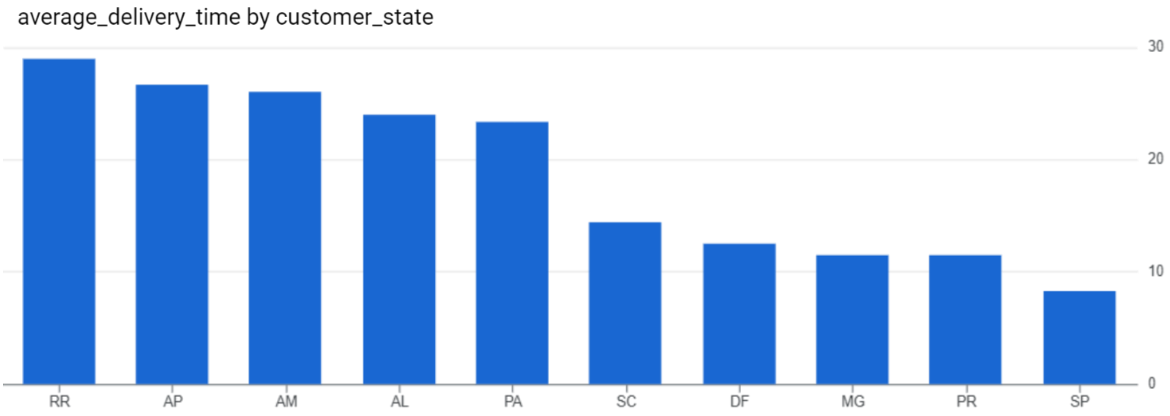
Ans:

```
with top as
(
SELECT
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,day)),2) AS average_delivery_time
FROM target.orders o
JOIN
target.customers c ON o.customer_id = c.customer_id
WHERE o.order_status = 'delivered'
GROUP BY customer_state
ORDER BY average_delivery_time DESC
LIMIT 5
),
bottom as
(
SELECT
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,day)),2) AS average_delivery_time
FROM target.orders o
JOIN
target.customers c ON o.customer_id = c.customer_id
WHERE o.order_status = 'delivered'
GROUP BY customer_state
ORDER BY average_delivery_time ASC
LIMIT 5
)
select * from top
union distinct
select * from bottom
order by 2 desc
```



Row	customer_state	average_delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32
6	SC	14.48
7	DF	12.51
8	MG	11.54
9	PR	11.53
10	SP	8.3

Chart:



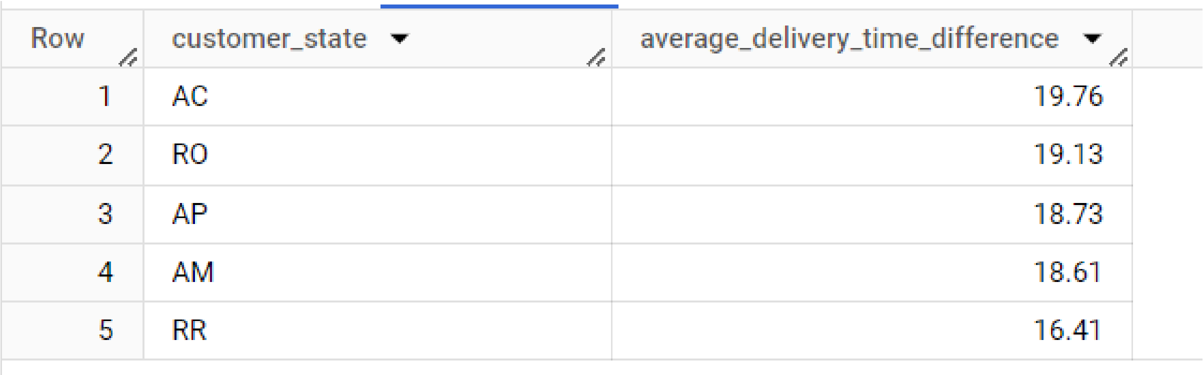
Insights:

1. States like RR, AP, AM are taking highest delivery time for an order.
2. States like SP, PR, MG are taking lowest delivery time for an order.

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

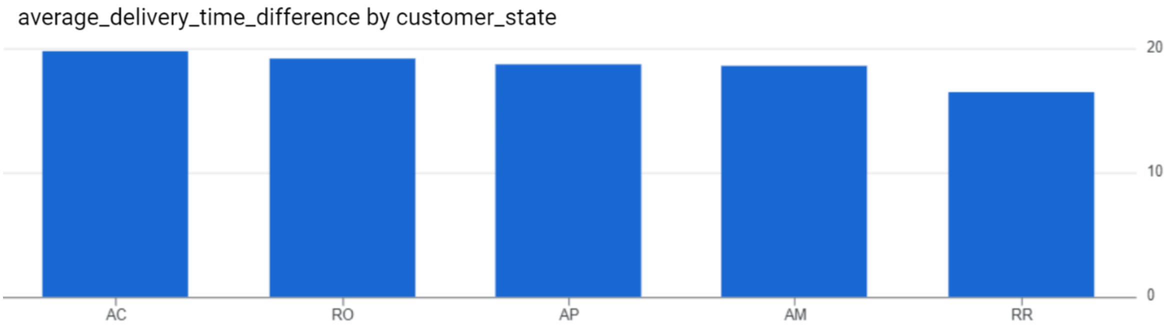
Ans:

```
SELECT
c.customer_state,
round(avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day
)),2) AS average_delivery_time_difference
FROM target.orders o
JOIN
target.customers c ON o.customer_id = c.customer_id
WHERE o.order_status = 'delivered'
GROUP BY customer_state
ORDER BY average_delivery_time_difference DESC
LIMIT 5
```



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

Chart:



Insights:

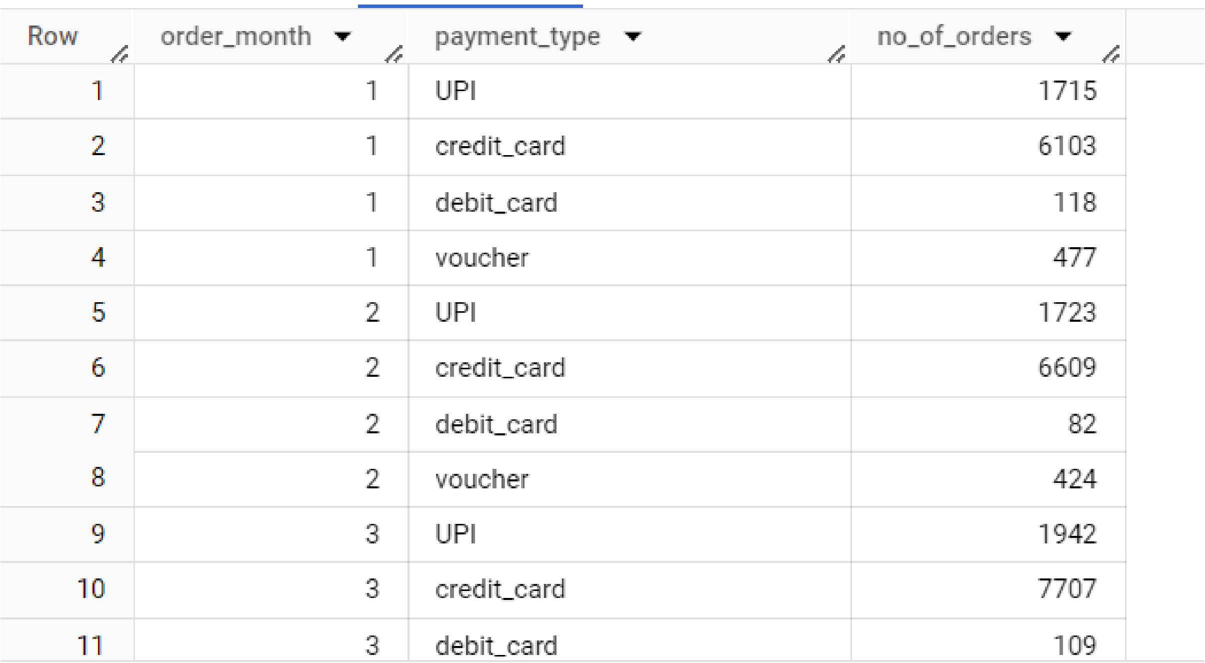
1. The top 5 states with the fastest order deliveries compared to the estimated dates are AC,RO ,AP ,AM and RR .
2. Faster deliveries in these states could contribute to higher customer satisfaction, as customers receive their orders sooner than expected.

VI. Analysis based on the payments:

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

Ans:

```
select extract(month from order_purchase_timestamp) as
order_month,payment_type,count(*) as no_of_orders
from target.orders o
JOIN target.payments p using (order_id)
group by order_month,p.payment_type
order by 1,2
```



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

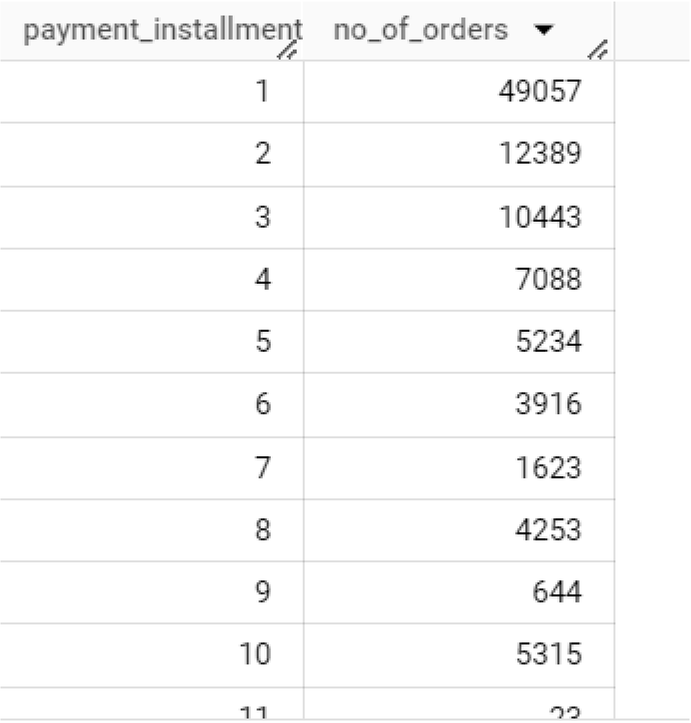
Insights:

1. Payment types like "UPI," "credit_card," and "voucher" show relatively consistent patterns throughout the year. They consistently have a higher number of orders.
2. The number of orders placed using different payment types shows variation across different months.
3. "Debit_card" and "not_defined" payment types have significantly fewer orders compared to others, and they also exhibit variations over the months.
4. It's interesting to note that during the holiday season (e.g., in December), some payment types like "voucher" and "credit_card" have a spike in the number of orders

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

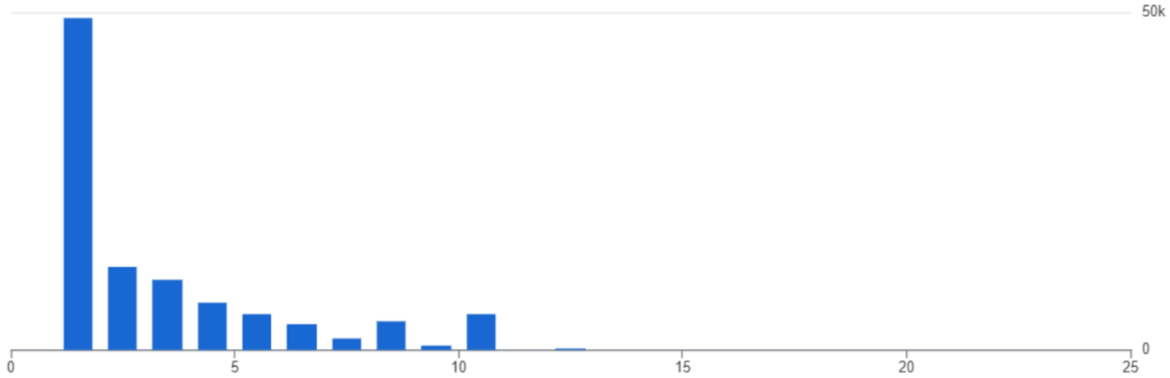
Ans:

```
SELECT
    payment_installments ,
    COUNT(DISTINCT order_id) AS no_of_orders
FROM
    target.payments
WHERE
    payment_value > 0 and payment_installments>0
GROUP BY
    payment_installments
ORDER BY
    payment_installments;
```



payment_installment	no_of_orders ▼
1	49057
2	12389
3	10443
4	7088
5	5234
6	3916
7	1623
8	4253
9	644
10	5315
11	23

no_of_orders by payment_installments



Insights:

1. Most orders have 1 payment installment, with 49,057 orders falling into this category.

2. The number of orders decreases as the number of payment installments increases, with a noticeable drop after 10 installments.

3. There are very few orders with a high number of payment installments (e.g., 21, 22, 23, 24), indicating that most customers opt for a smaller number of installments.

4. Orders with 11, 13, 14, 16, and 17 payment installments are relatively rare.