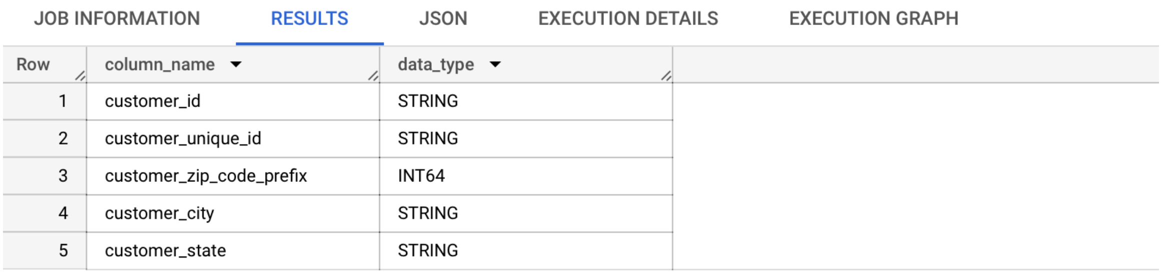
Actionable Insights & Recommendations - **At the end**

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

1. Data type of all columns in the "customers" table.

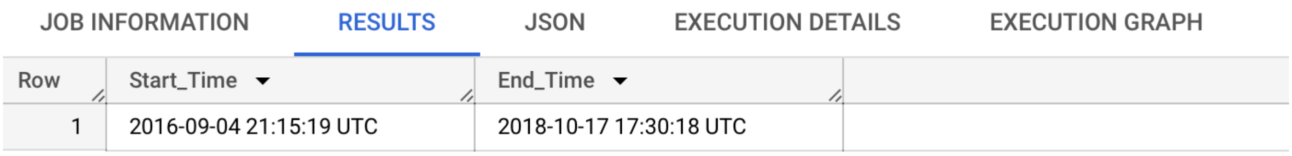
ANS -
```
SELECT
    column_name,
    data_type
    FROM
    `Target.INFORMATION_SCHEMA.COLUMNS`
    WHERE
    table_name = ‘customers’
```



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

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

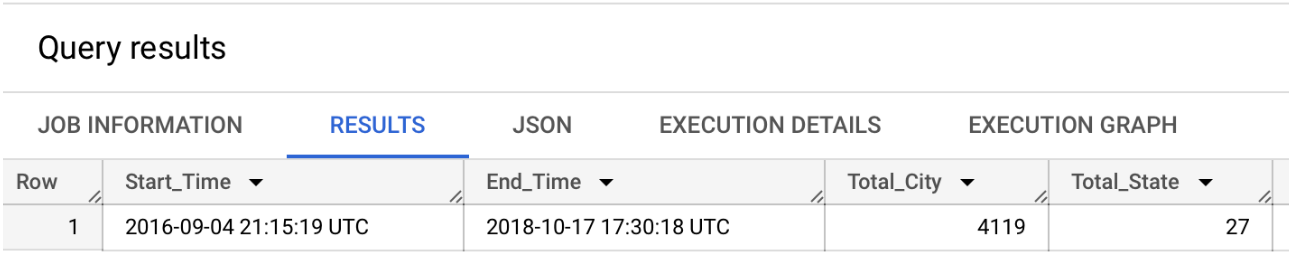
ANS -
```
SELECT MIN(order_purchase_timestamp) as Start_Time,
    MAX(order_purchase_timestamp) as End_Time
    FROM `Target.orders`
```



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

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

ANS -
```
SELECT MIN(o.order_purchase_timestamp) AS Start_Time,
    MAX(o.order_purchase_timestamp) AS End_Time,
    COUNT(Distinct c.customer_city) AS Total_City,
    COUNT(Distinct c.customer_state) AS Total_State
    FROM `Target.orders` o JOIN `Target.customers` c
    ON  o.customer_id = c.customer_id
```

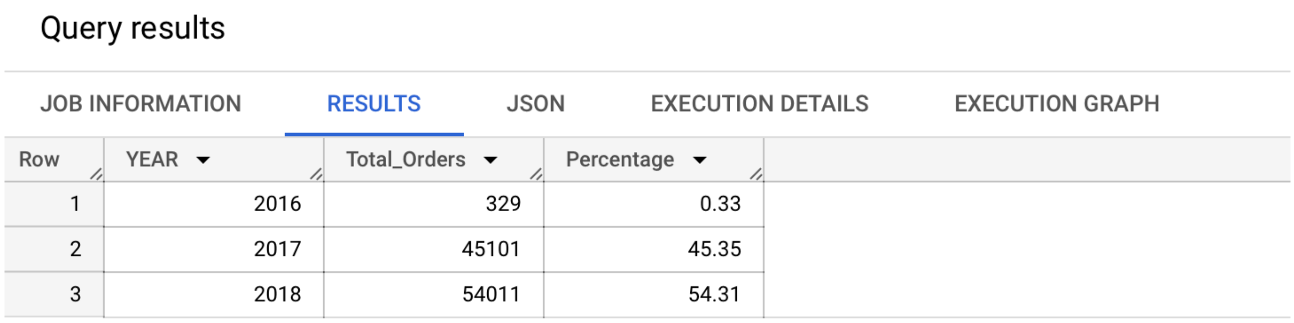

Query results

Row	Start_Time	End_Time	Total_City	Total_State
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	4119	27

2. In-depth Exploration:

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

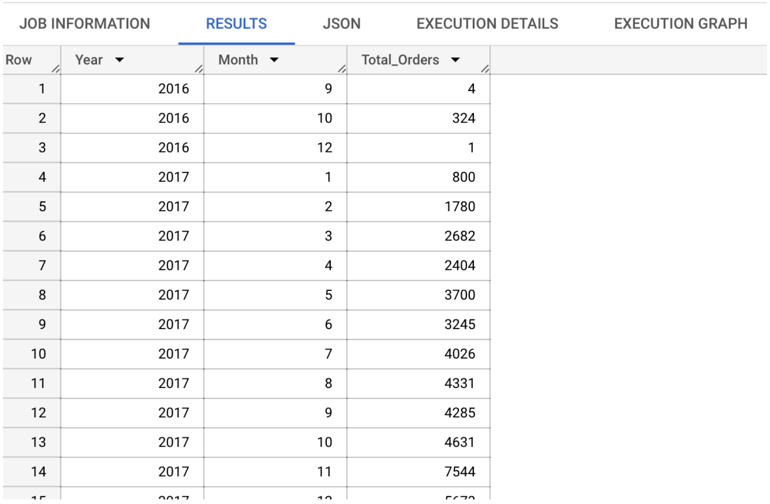
ANS -
```
SELECT extract(YEAR from order_purchase_timestamp) as YEAR,
count(distinct order_id) as Total_Orders,
ROUND(COUNT(order_id) * 100.0 / SUM(COUNT(order_id)) OVER(),2) AS Percentage
FROM `Target.orders`
GROUP BY YEAR
ORDER BY YEAR
```


Query results

Row	YEAR	Total_Orders	Percentage
1	2016	329	0.33
2	2017	45101	45.35
3	2018	54011	54.31

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

ANS -
```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS Year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS Month,
COUNT(DISTINCT order_id) AS Total_Orders
FROM `Target.orders`
GROUP BY Year, Month ORDER BY Year, Month;
```



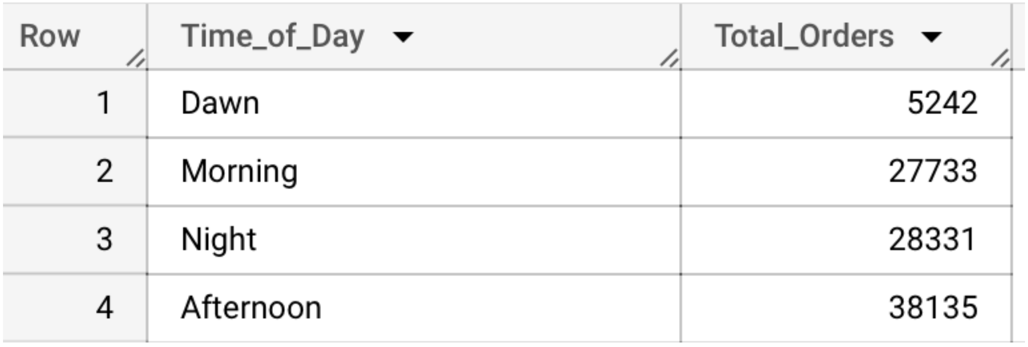
Row	Year	Month	Total_Orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673

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

ANS -

```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN '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'
    ELSE 'Other'
  END AS Time_of_Day,
COUNT(DISTINCT order_id) AS Total_Orders
FROM `Target.orders`
WHERE order_purchase_timestamp IS NOT NULL
GROUP BY Time_of_Day
ORDER BY Total_Orders
```



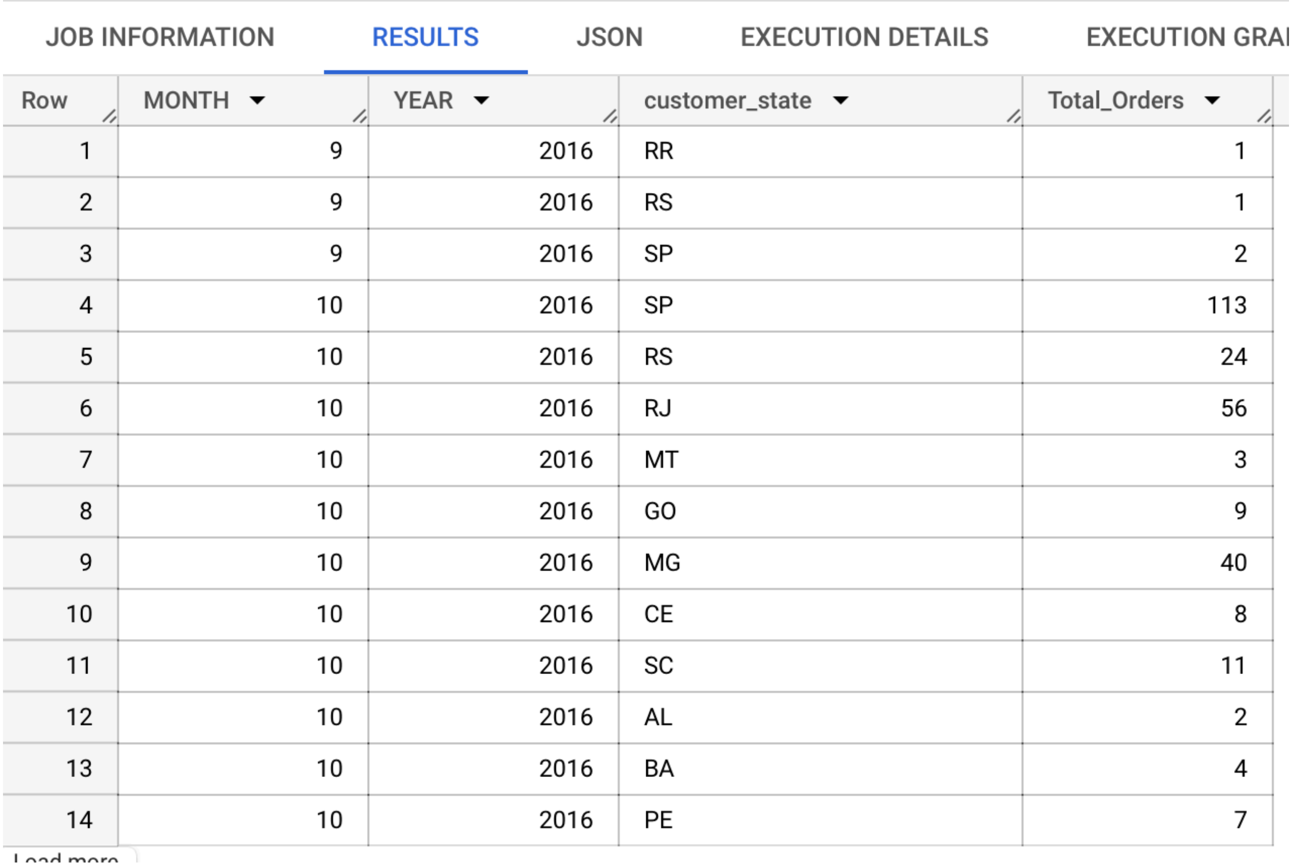
Row	Time_of_Day	Total_Orders
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

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

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

ANS

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) as MONTH,
EXTRACT(YEAR FROM order_purchase_timestamp) as YEAR,
customer_state,
COUNT(DISTINCT order_id) as Total_Orders
FROM `Target.orders` o JOIN `Target.customers` c
ON o.customer_id = c.customer_id
GROUP BY MONTH, YEAR, customer_state
ORDER BY YEAR, MONTH
```

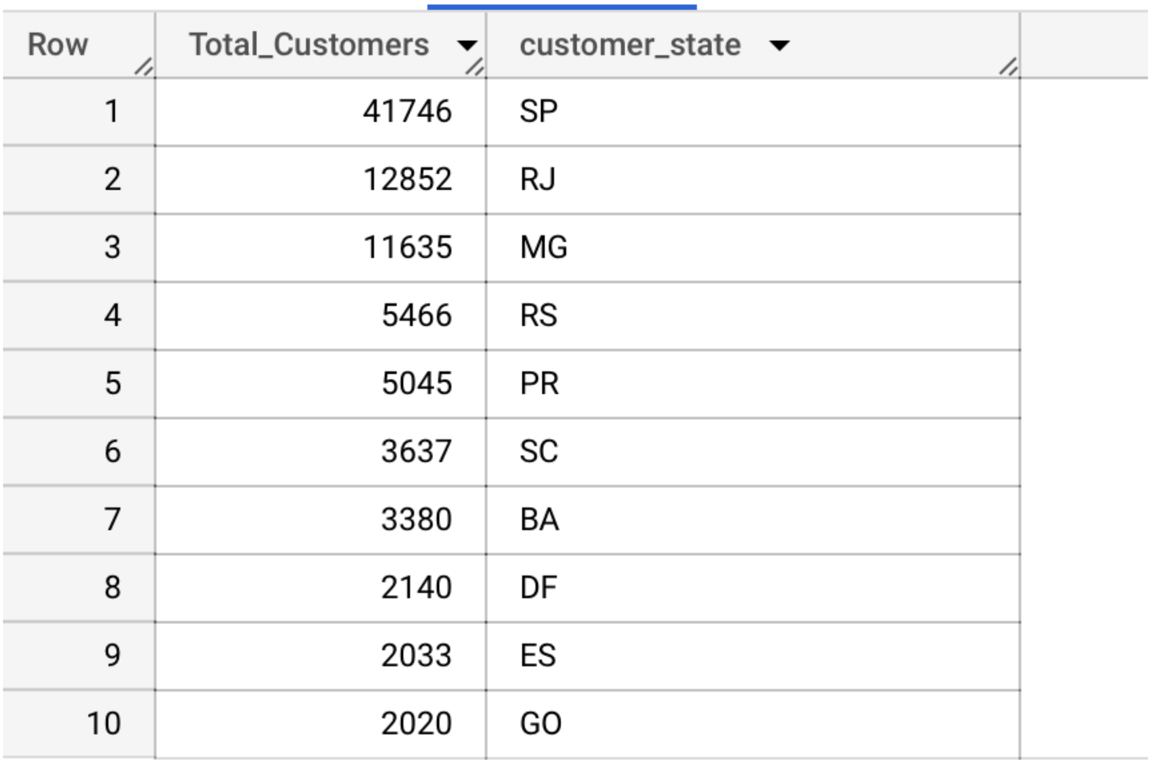


Row	MONTH	YEAR	customer_state	Total_Orders
1	9	2016	RR	1
2	9	2016	RS	1
3	9	2016	SP	2
4	10	2016	SP	113
5	10	2016	RS	24
6	10	2016	RJ	56
7	10	2016	MT	3
8	10	2016	GO	9
9	10	2016	MG	40
10	10	2016	CE	8
11	10	2016	SC	11
12	10	2016	AL	2
13	10	2016	BA	4
14	10	2016	PE	7

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

ANS-

```
SELECT count(customer_id) as Total_Customers,
customer_state
FROM `Target.customers`
GROUP BY customer_state
ORDER BY Total_Customers DESC
LIMIT 10
```



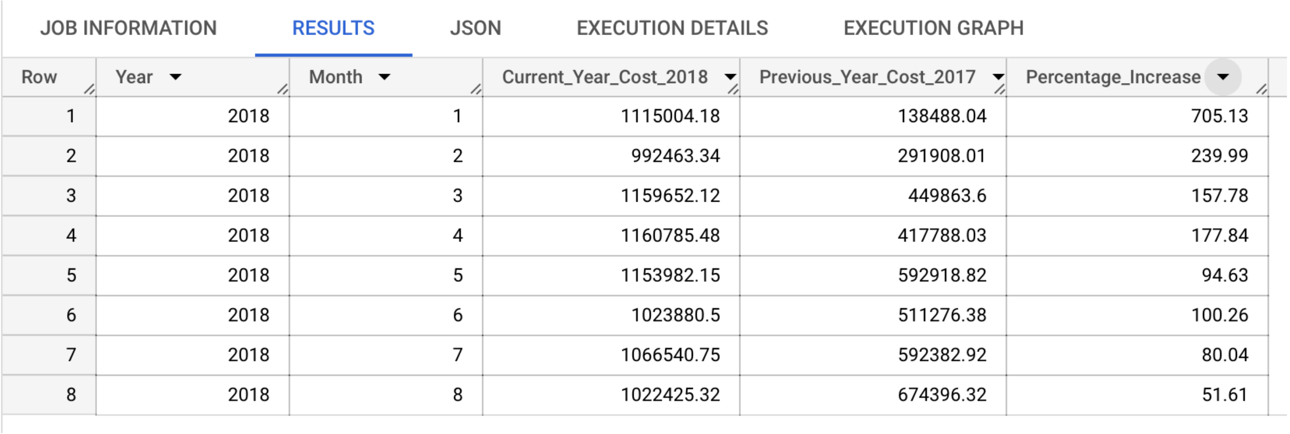
Row	Total_Customers	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO

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.

ANS -

```
WITH YearMonthCost AS (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS Year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS Month,
    ROUND(SUM(p.payment_value),2) AS Total_Cost
  FROM
    `Target.payments` p
  JOIN
    `Target.orders` o ON p.order_id = o.order_id
  WHERE
    EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018)
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
  GROUP BY
    Year,
    Month
),
Year2017 AS (SELECT Year,  Month, Total_Cost FROM YearMonthCost
 WHERE Year = 2017
),
Year2018 AS (SELECT Year, Month, Total_Cost FROM YearMonthCost
WHERE Year = 2018
)
SELECT
  y2018.Year,
  y2018.Month,
  y2018.Total_Cost AS Current_Year_Cost_2018,
  y2017.Total_Cost AS Previous_Year_Cost_2017,
  ROUND(((y2018.Total_Cost - y2017.Total_Cost) / y2017.Total_Cost) * 100,2) AS
Percentage_Increase
FROM Year2018 y2018
JOIN Year2017 y2017 ON y2018.Month = y2017.Month
ORDER BY y2018.Year, y2018.Month;
```

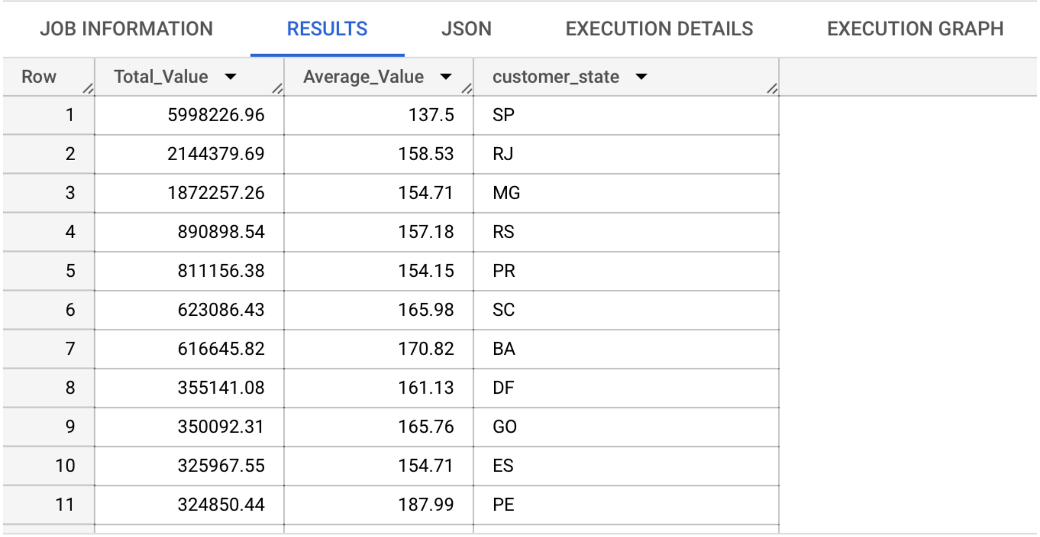


Row	Year	Month	Current_Year_Cost_2018	Previous_Year_Cost_2017	Percentage_Increase
1	2018	1	1115004.18	138488.04	705.13
2	2018	2	992463.34	291908.01	239.99
3	2018	3	1159652.12	449863.6	157.78
4	2018	4	1160785.48	417788.03	177.84
5	2018	5	1153982.15	592918.82	94.63
6	2018	6	1023880.5	511276.38	100.26
7	2018	7	1066540.75	592382.92	80.04
8	2018	8	1022425.32	674396.32	51.61

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

ANS -

```
SELECT ROUND(sum(p.payment_value),2) as Total_Value,
ROUND(AVG(p.payment_value),2) as Average_Value,
c.customer_state
FROM `Target.payments` p JOIN `Target.orders` o
ON p.order_id = o.order_id
JOIN `Target.customers` c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Total_Value DESC
```

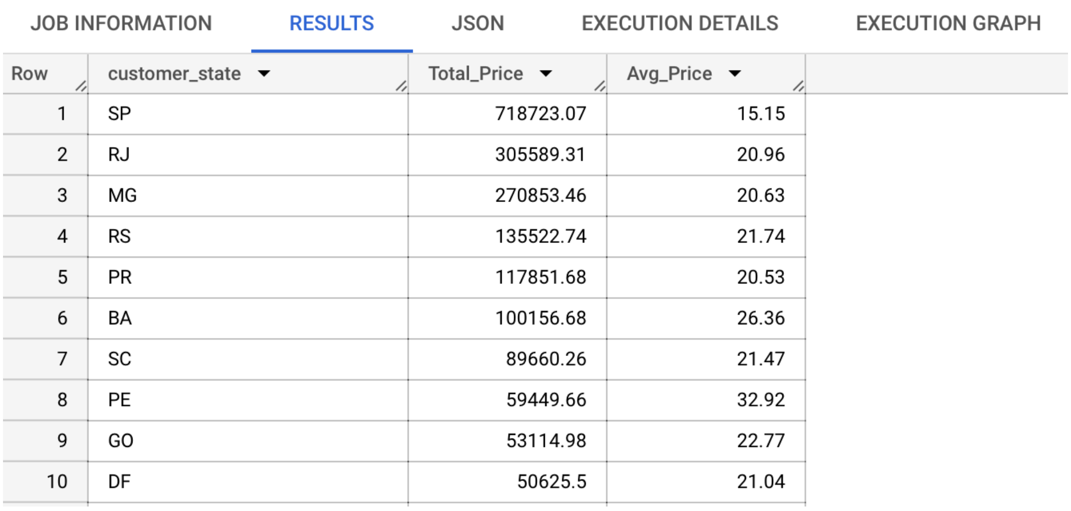


Row	Total_Value	Average_Value	customer_state
1	5998226.96	137.5	SP
2	2144379.69	158.53	RJ
3	1872257.26	154.71	MG
4	890898.54	157.18	RS
5	811156.38	154.15	PR
6	623086.43	165.98	SC
7	616645.82	170.82	BA
8	355141.08	161.13	DF
9	350092.31	165.76	GO
10	325967.55	154.71	ES
11	324850.44	187.99	PE

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

ANS -

```
SELECT c.customer_state, ROUND(SUM(o.freight_value),2)AS Total_Price,
 ROUND(AVG(o.freight_value),2) AS Avg_Price
 FROM `Target.order_items` o JOIN `Target.orders` oh ON o.order_id = oh.order_id
JOIN `Target.customers` c ON oh.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Total_Price DESC
```



Row	customer_state	Total_Price	Avg_Price
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

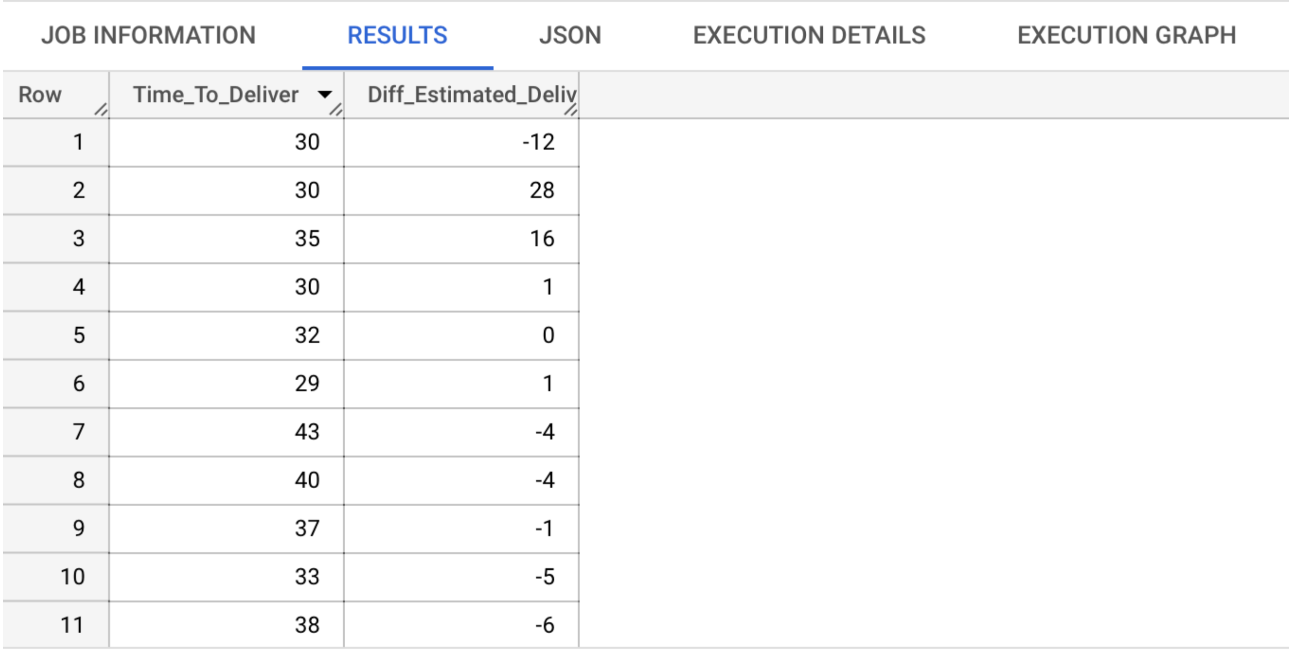
Analysis based on sales, freight and delivery time.

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

ANS-

```
SELECT
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.orders`
```

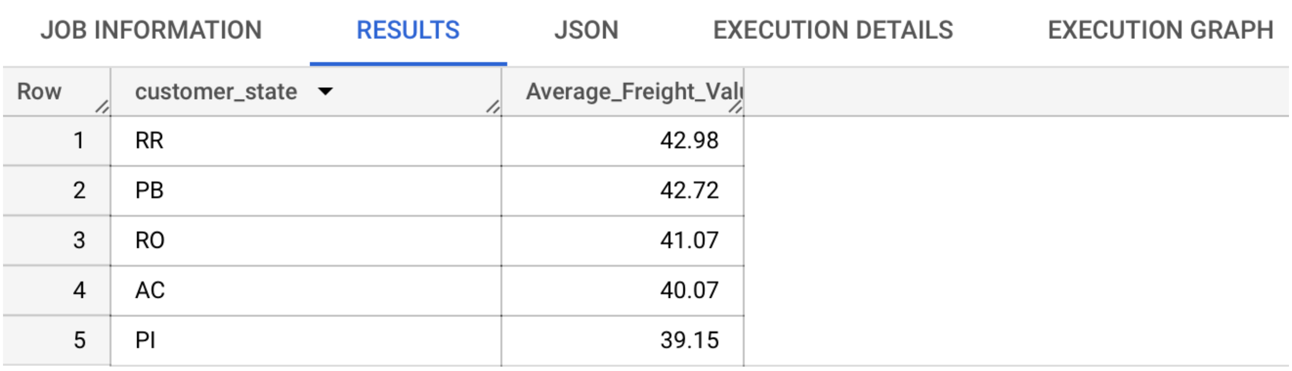


Row	Time_To_Deliver	Diff_Estimated_Deliv
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	29	1
7	43	-4
8	40	-4
9	37	-1
10	33	-5
11	38	-6

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

ANS-

```
SELECT c.customer_state, ROUND(AVG(o.freight_value),2)AS Average_Freight_Value,
 FROM `Target.order_items` o JOIN `Target.orders` oh ON o.order_id = oh.order_id
JOIN `Target.customers` c ON oh.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Average_Freight_Value DESC
LIMIT 5
```

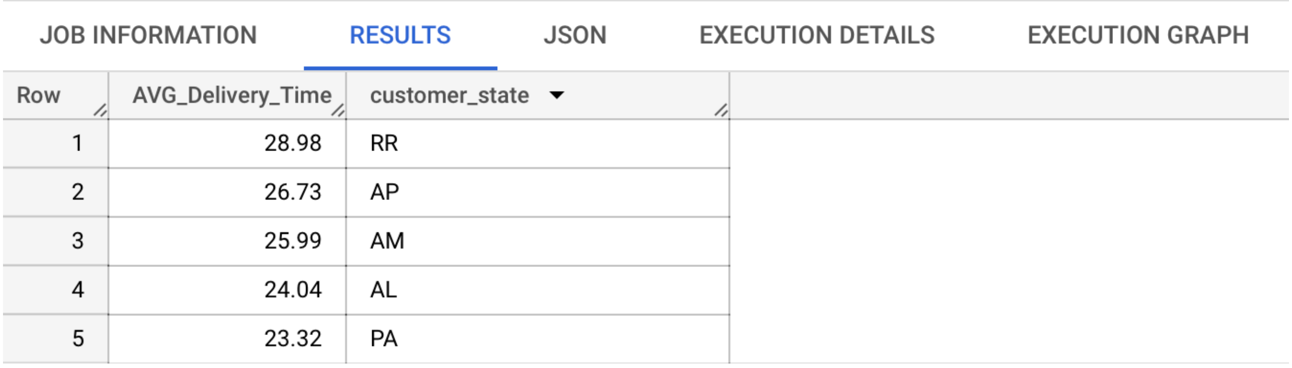


Row	customer_state	Average_Freight_Val
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

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

ANS-

```
SELECT
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)),2) as
AVG_Delivery_Time,
c.customer_state
FROM `Target.orders` o JOIN `Target.customers` c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY AVG_Delivery_Time DESC
LIMIT 5
```

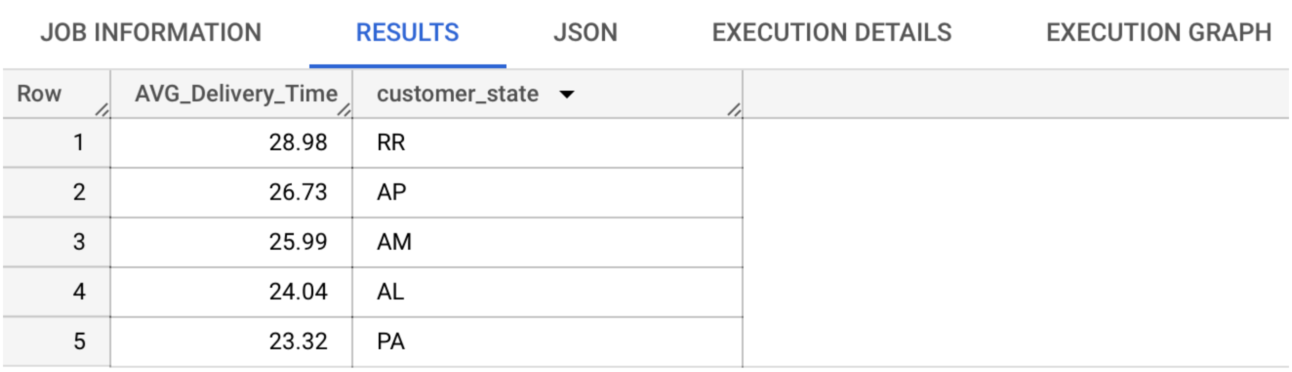


Row	AVG_Delivery_Time	customer_state
1	28.98	RR
2	26.73	AP
3	25.99	AM
4	24.04	AL
5	23.32	PA

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

ANS -

```
SELECT
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)),2) as AVG_Delivery_Time, customer_state
FROM `Target.orders` o JOIN `Target.customers` c
ON o.customer_id = c.customer_id
WHERE order_status='delivered'
GROUP BY c.customer_state
HAVING AVG_Delivery_Time > 0
ORDER BY AVG_Delivery_Time DESC
LIMIT 5
```



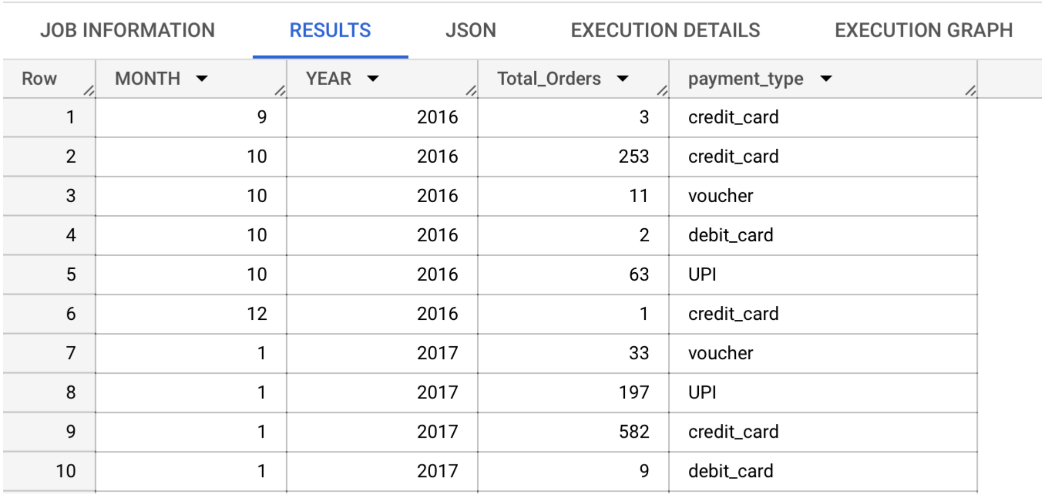
Row	AVG_Delivery_Time	customer_state
1	28.98	RR
2	26.73	AP
3	25.99	AM
4	24.04	AL
5	23.32	PA

Analysis based on the payments:

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

ANS -

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) as MONTH,
EXTRACT(YEAR FROM order_purchase_timestamp) as YEAR,
COUNT(DISTINCT o.order_id) as Total_Orders,
p.payment_type,
FROM `Target.orders` o JOIN `Target.payments` p
ON o.order_id = p.order_id
GROUP BY MONTH, YEAR, p.payment_type
ORDER BY YEAR, MONTH
```

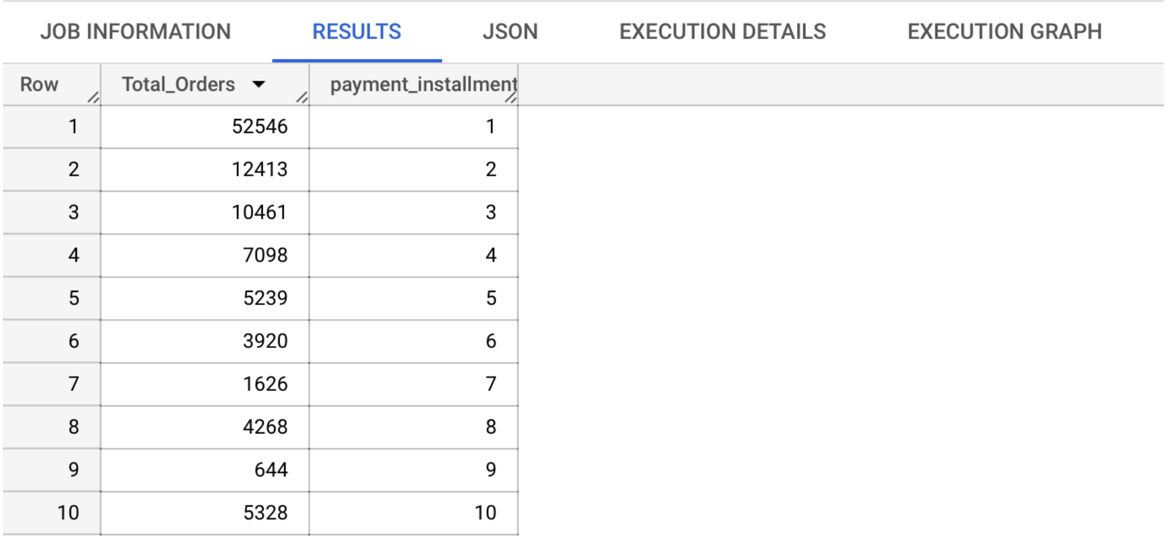


Row	MONTH	YEAR	Total_Orders	payment_type
1	9	2016	3	credit_card
2	10	2016	253	credit_card
3	10	2016	11	voucher
4	10	2016	2	debit_card
5	10	2016	63	UPI
6	12	2016	1	credit_card
7	1	2017	33	voucher
8	1	2017	197	UPI
9	1	2017	582	credit_card
10	1	2017	9	debit_card

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

ANS-

```
SELECT count(order_id) AS Total_Orders,
payment_installments
FROM `Target.payments`
GROUP BY payment_installments
HAVING payment_installments > 0
ORDER BY payment_installments
```



Row	Total_Orders	payment_installment
1	52546	1
2	12413	2
3	10461	3
4	7098	4
5	5239	5
6	3920	6
7	1626	7
8	4268	8
9	644	9
10	5328	10

Actionable Insights & Recommendations (10 points) Based on the data analysis conducted so far, here are some actionable insights and recommendations-

PAYMENT TYPE PREFERENCES ->
Based on the month-on-month analysis, we can observe that the majority of orders are placed using credit cards, followed by UPI, and then other payment types. This trend indicates that credit cards are the most preferred payment method among customers, with UPI gaining popularity as the second most used payment option. While other payment types also contribute to the overall orders, they show comparatively lower usage in comparison to credit cards and UPI.
It is important for businesses to focus on optimising the payment experience for credit card and UPI users to ensure a seamless and efficient checkout process. Additionally, continuous monitoring and analysis of customer payment preferences can help identify any shifts in behaviour or the emergence of new payment options, allowing businesses to adapt their payment processing strategies accordingly.

ORDER PATTERN OF THE CUSTOMER ->
Based on the analysis of the time of day when most orders are placed by customers, we can observe the following patterns:

- **Afternoon**: The highest number of orders are placed during the afternoon hours. This suggests that customers are actively making purchases during this time, possibly after completing their daily activities and having some leisure time to shop.
- **Night**: The second most popular time for order placements is during the night. This indicates that many customers prefer to shop online in the evening or late hours, possibly after work or other commitments.
- **Morning**: The morning hours rank third in terms of order placements. It suggests that some customers may start their day by making purchases online or planning their purchases ahead of time.
- **Dawn**: The least number of orders are placed during the dawn hours. This time of day might have relatively fewer customers making purchases, possibly due to it being early in the morning.
-

GROWING TREND IN THE NO. OF ORDERS PLACED OVER THE PAST YEARS ->
There is a notable growth in the number of orders placed each year. In 2016, there were 329 orders, which increased significantly to 45,101 orders in 2017, indicating substantial growth in customer demand. However, it's worth noting that there seems to be a data loss or incomplete data for 2017, as such a significant jump in orders is unexpected and may be attributed to data inconsistencies or missing records.
Despite the data challenges in 2017, the growth trend continued in 2018, with the number of orders reaching 54,011. This further validates the positive trajectory of the business and indicates that customer engagement and sales performance improved over time.

SOME SIGNIFICANT TRENDS IN THE NUMBER OF ORDERS PLACED->

1. High Orders in November 2017: November 2017 stands out as a month with exceptionally high order volume. This could be attributed to various factors, such as festive season sales, Black Friday deals, or promotional campaigns that attracted a large number of customers.

2. Consistent Growth from January 2017: Starting from January 2017, we observe a consistent increase in the number of orders each month. This upward trend suggests a positive and steady growth in customer demand and overall business performance over time.

3. Data Discrepancy in December 2016: There appears to be a data discrepancy in December 2016, as the number of orders for that month is significantly lower compared to December 2017.

This anomaly could be due to incomplete or missing data for December 2016, leading to an inaccurate representation of the actual order volume.

To ensure accurate analysis and informed decision-making, it is crucial to address the data inconsistency in December 2016 and investigate the reasons behind the data loss. Rectifying data discrepancies will provide a more reliable picture of business performance during that period.

ORDERS PLACED ON THE BASIS OF THE PAYMENT INSTALLMENTS->

Orders with 1 Instalment: The highest number of orders, amounting to 52,546, are placed with a single instalment. This suggests that a significant portion of customers prefers to make their payments in a single transaction, possibly due to convenience or simplicity.

Decreasing Orders with Increasing Installments: As the number of installments increases, the number of orders decreases. For instance, there are 644 orders with 9 installments and 632 orders with 10 installments. This trend indicates that a smaller proportion of customers opt for paying in multiple installments, possibly due to factors such as interest charges or the desire for faster payment completion.