Business case study

TARGET

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.

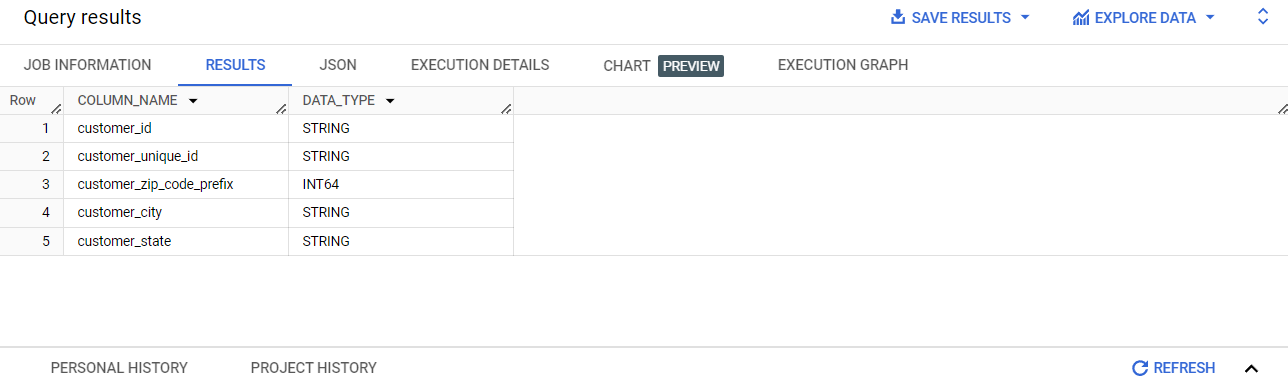
Query

SELECT COLUMN\_NAME, DATA\_TYPE

FROM target.INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = "customers"

Output



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

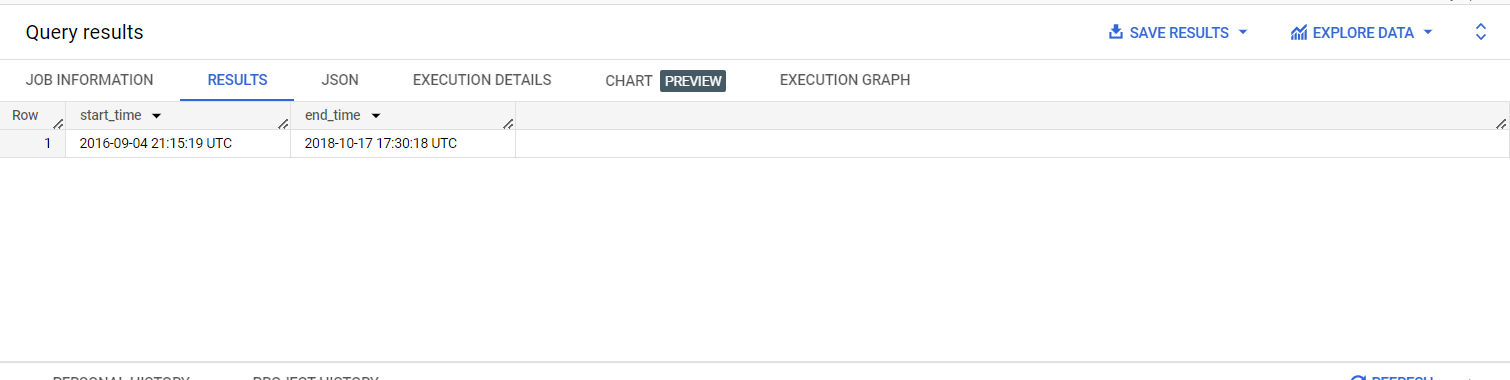
query:

SELECT MIN(order\_purchase\_timestamp) AS start\_time,

MAX(order\_purchase\_timestamp) AS end\_time

FROM `ecommerce-399716.target.orders`

Output:



**Insights**- most no.of orders placed in the year 2016- 2018 and timing 17:30 - 21:15

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

Query :

SELECT COUNT(DISTINCT C.customer\_city) AS Cities, COUNT (DISTINCT C.customer\_state) AS States

FROM `ecommerce-399716.target.customers` AS C

JOIN `ecommerce-399716.target.orders` AS O

ON C.customer\_id = O.customer\_id

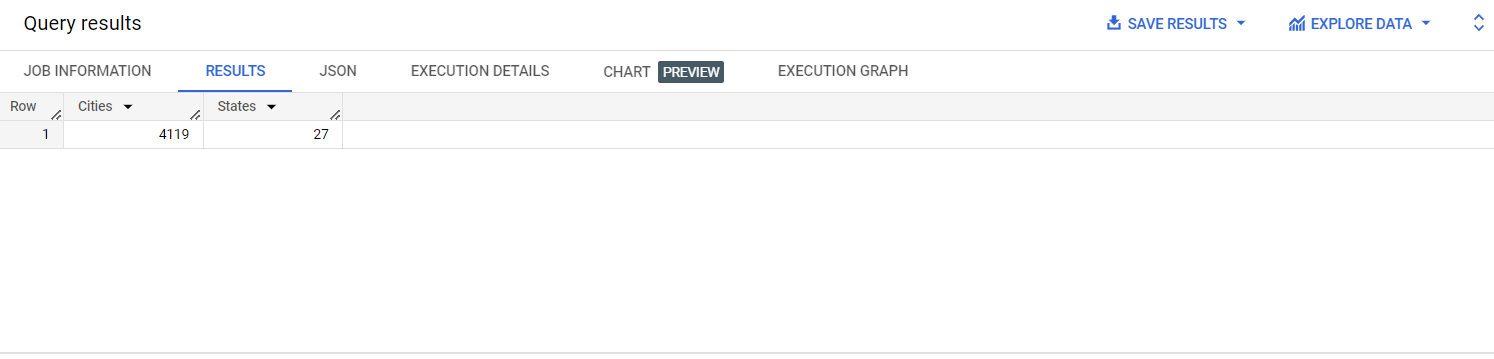
WHERE O.order\_purchase\_timestamp >=

"2016-09-04 21:15:19"

AND O.order\_purchase\_timestamp <=

"2018-10-17 17:30:18"

Output :



**Insight**- From 27 states and 4119 cities orders most of the orders were placed

**2. In-depth Exploration:**

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

Query:

SELECT EXTRACT( YEAR FROM order\_purchase\_timestamp) AS order\_year,

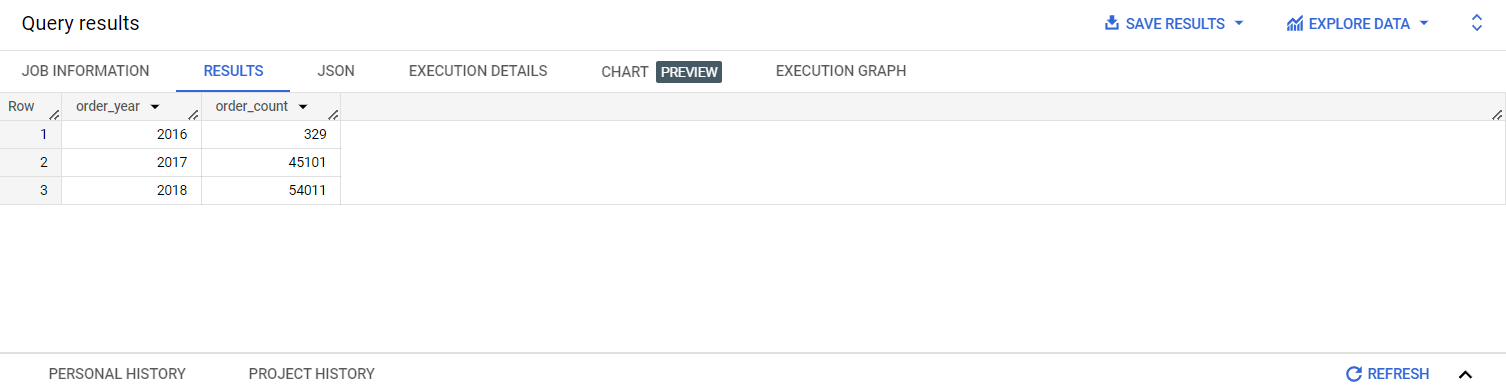
COUNT(\*) AS order\_count

FROM `ecommerce-399716.target.orders`

GROUP BY order\_year

ORDER BY order\_year ASC

Output:



Insights

Yes there is a growing trend in the year 2017, there was a high increase and 2017-2018 it was gradual .. if the pace of increase maintained as 2016- 2017 then company can expect higher percentage of profit

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

Query:

SELECT FORMAT\_DATETIME("%B", order\_purchase\_timestamp) AS order\_month,

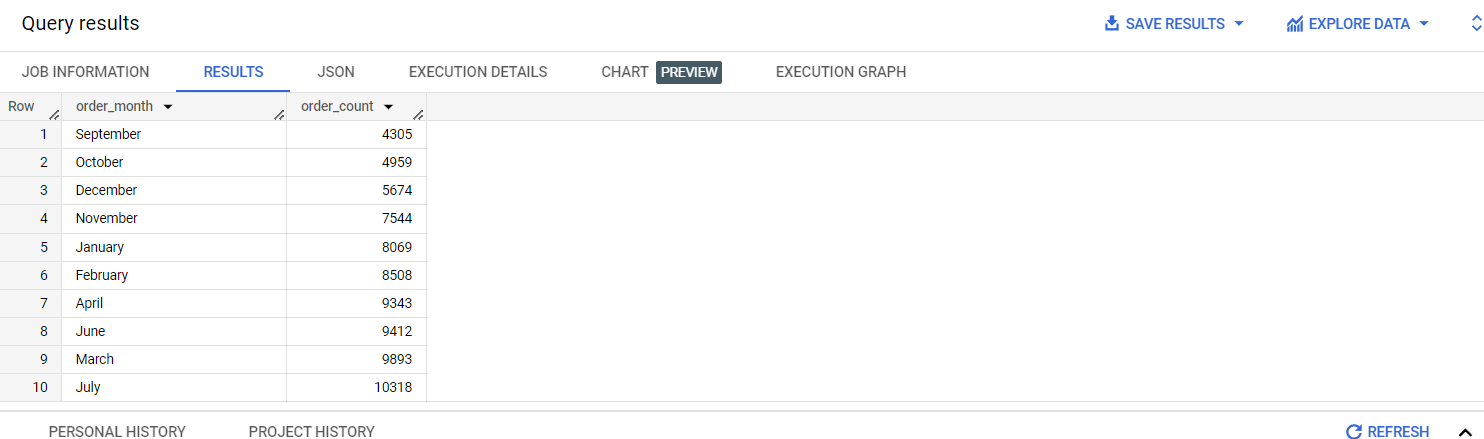
COUNT(\*) AS order\_count

FROM `ecommerce-399716.target.orders`

GROUP BY order\_month

ORDER BY order\_count ASC

Output :



Insights :

In the month of july there is higher number of orders , hence company can make use of it

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:

WITH OrderTimeSlots AS (

SELECT

EXTRACT(HOUR FROM order\_purchase\_timestamp) AS order\_hour,

COUNT(\*) AS order\_count

FROM `ecommerce-399716.target.orders`

GROUP BY

order\_hour

)

SELECT

CASE

WHEN order\_hour BETWEEN 0 AND 6 THEN 'Dawn'

WHEN order\_hour BETWEEN 7 AND 12 THEN 'Morning'

WHEN order\_hour BETWEEN 13 AND 18 THEN 'Afternoon'

WHEN order\_hour BETWEEN 19 AND 23 THEN 'Night'

ELSE NULL

END AS time\_slot,

SUM(order\_count) AS total\_orders

FROM

OrderTimeSlots

GROUP BY

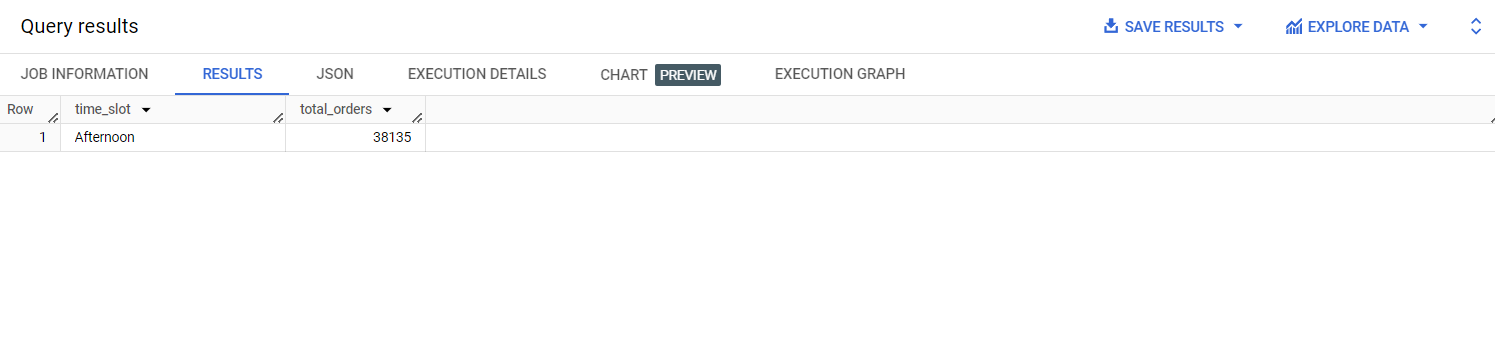
time\_slot

ORDER BY

total\_orders DESC

limit 1

Output:



**Insights**- Brazilian customers mostly place their orders 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 FORMAT\_DATETIME("%B", order\_purchase\_timestamp) AS order\_month,

COUNT(\*) AS order\_count, C.customer\_state

FROM `ecommerce-399716.target.orders` AS O

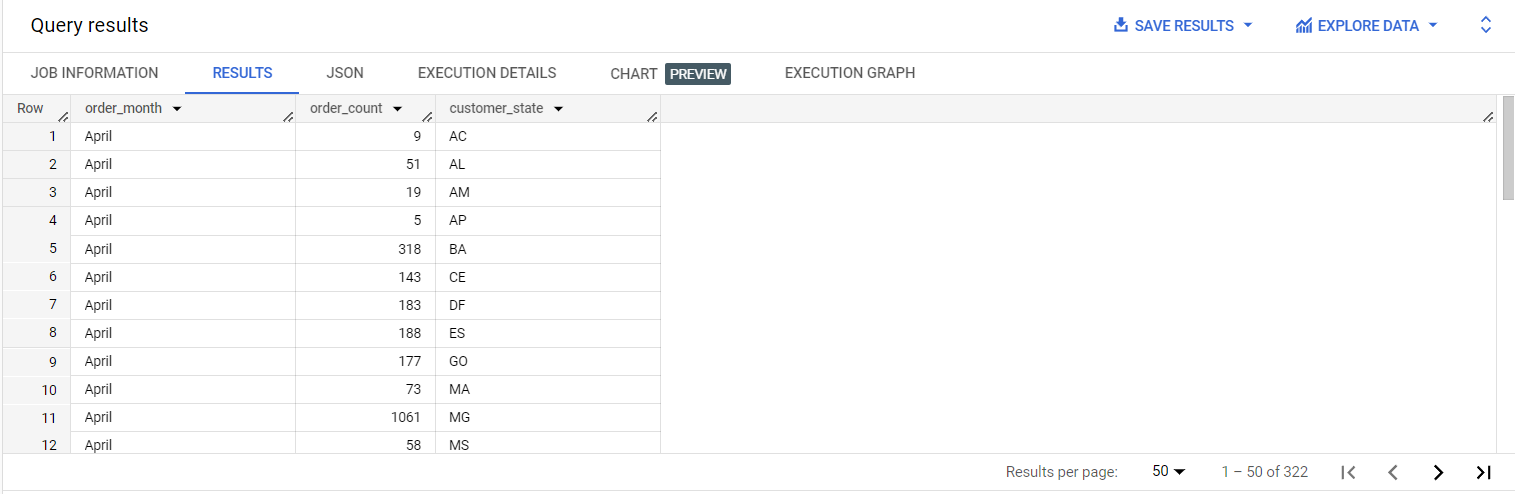
JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C.customer\_id

GROUP BY order\_month, C.customer\_state

ORDER BY order\_month, C.customer\_state ASC

Output:



**Insights**- higher number of orders placed in April, state MG has total order count 1061 which is highest

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

Query ;

SELECT customer\_state,

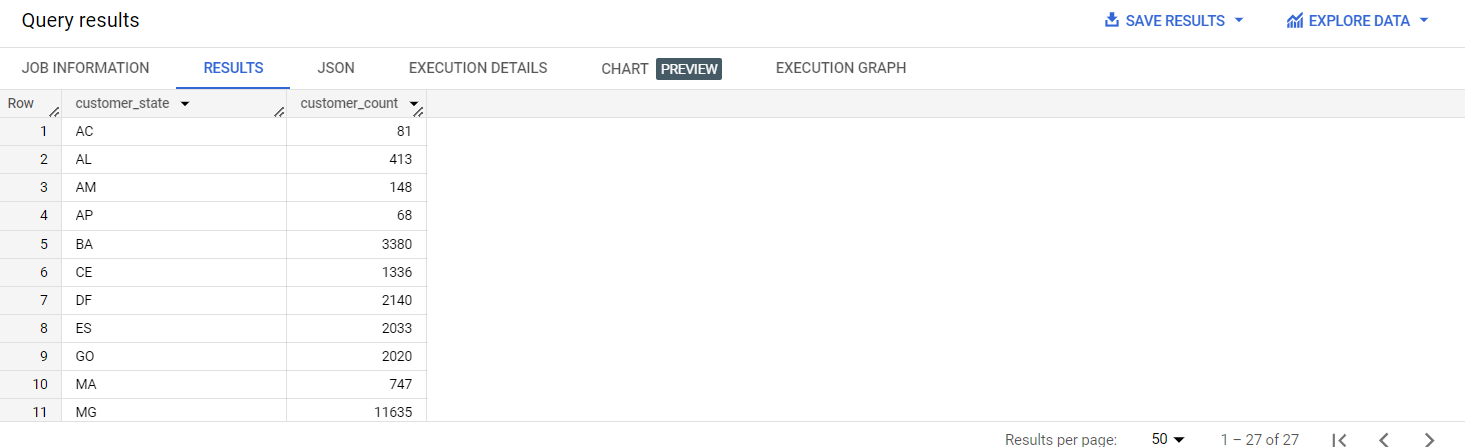
COUNT(\*) AS customer\_count

FROM `ecommerce-399716.target.customers`

GROUP BY customer\_state

ORDER BY customer\_state ASC

Output



**Insight**- high customer count is from MG state ie 11635 and lowest is from AP state that is 68

**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

WITH jan\_to\_aug AS

(SELECT SUM(CASE WHEN EXTRACT(YEAR FROM O.order\_purchase\_timestamp) = 2017 THEN P.payment\_value ELSE 0 END) AS total\_payment\_2017,

SUM(CASE WHEN EXTRACT(YEAR FROM O.order\_purchase\_timestamp) = 2018 THEN P.payment\_value ELSE 0 END) AS total\_payment\_2018

FROM `ecommerce-399716.target.orders` AS O

JOIN `ecommerce-399716.target.payments` AS P

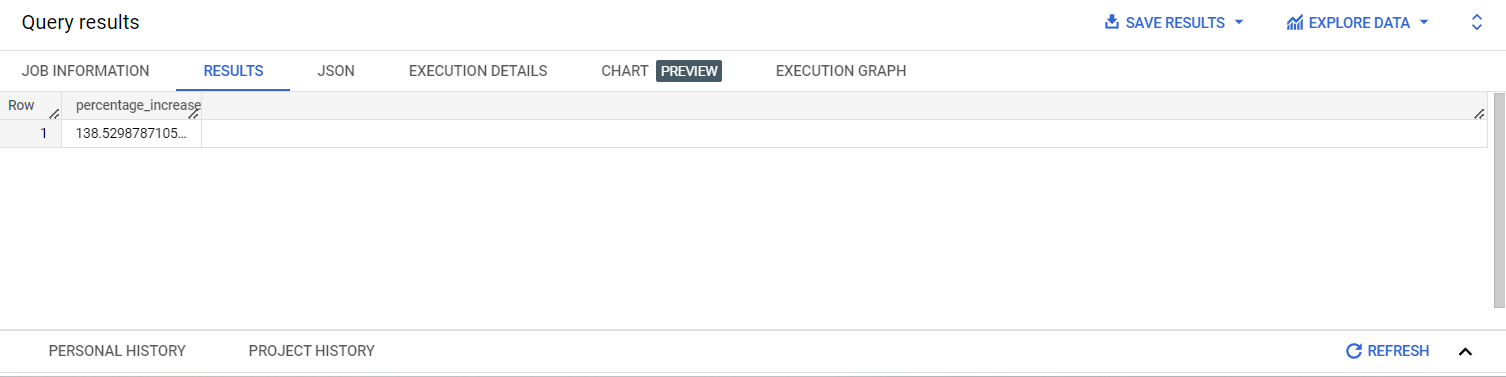
ON O.order\_id = P.order\_id

WHERE O.order\_purchase\_timestamp BETWEEN "2017-01-01" AND "2017-08-31" OR O.order\_purchase\_timestamp BETWEEN "2018-01-01" AND "2018-08-31")

SELECT ((jan\_to\_aug.total\_payment\_2018 - jan\_to\_aug.total\_payment\_2017) / jan\_to\_aug.total\_payment\_2017)\*100 AS percentage\_increase

FROM jan\_to\_aug

Output



**Insights**- average percentage increase is 138.52% from 2017-2018 january to august

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

Query:

SELECT C.customer\_state,

ROUND(SUM(P.payment\_value),2) AS total,

ROUND(AVG(P.payment\_value),2) AS avg

FROM `ecommerce-399716.target.orders` AS O

JOIN `ecommerce-399716.target.payments`AS P

ON O.order\_id = P.order\_id

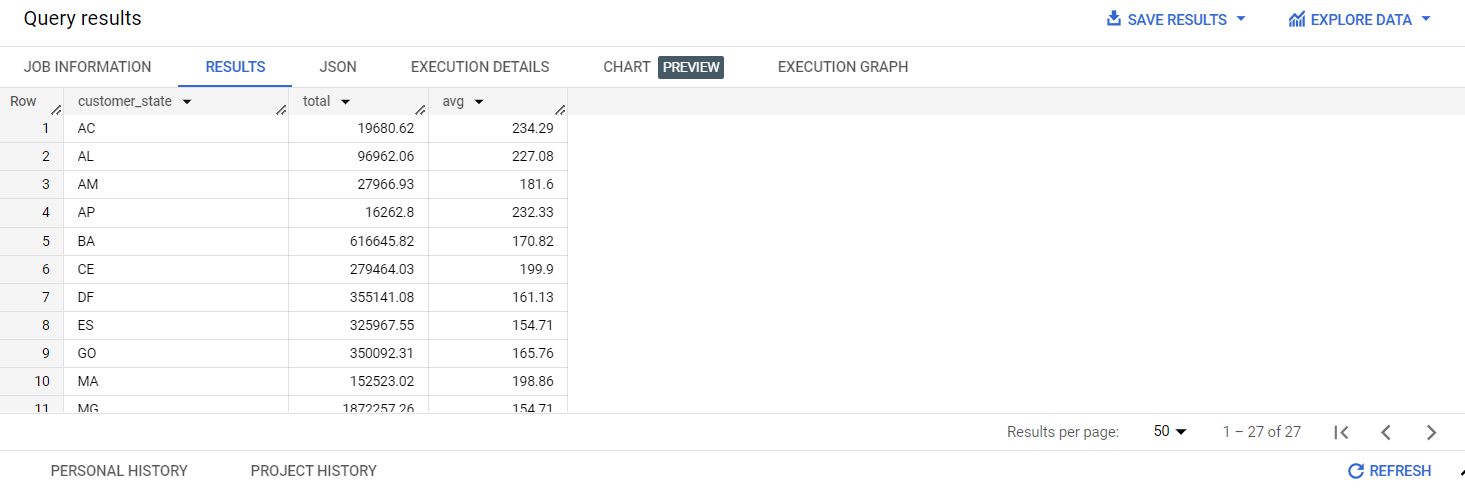
JOIN `ecommerce-399716.target.customers`AS C

ON O.customer\_id = C.customer\_id

GROUP BY C.customer\_state

ORDER BY C.customer\_state ASC

Output



**Insights-** state BA has highest total order price that is 616645.82 and highest average 198.96 from state MA

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

Query

SELECT C.customer\_state,

ROUND(SUM(freight\_value),2) AS total\_freight\_value,

ROUND(AVG(freight\_value),2) AS avg\_freight\_value

FROM `ecommerce-399716.target.order\_items` AS OT

JOIN `ecommerce-399716.target.orders` AS O

ON OT.order\_id = O.order\_id

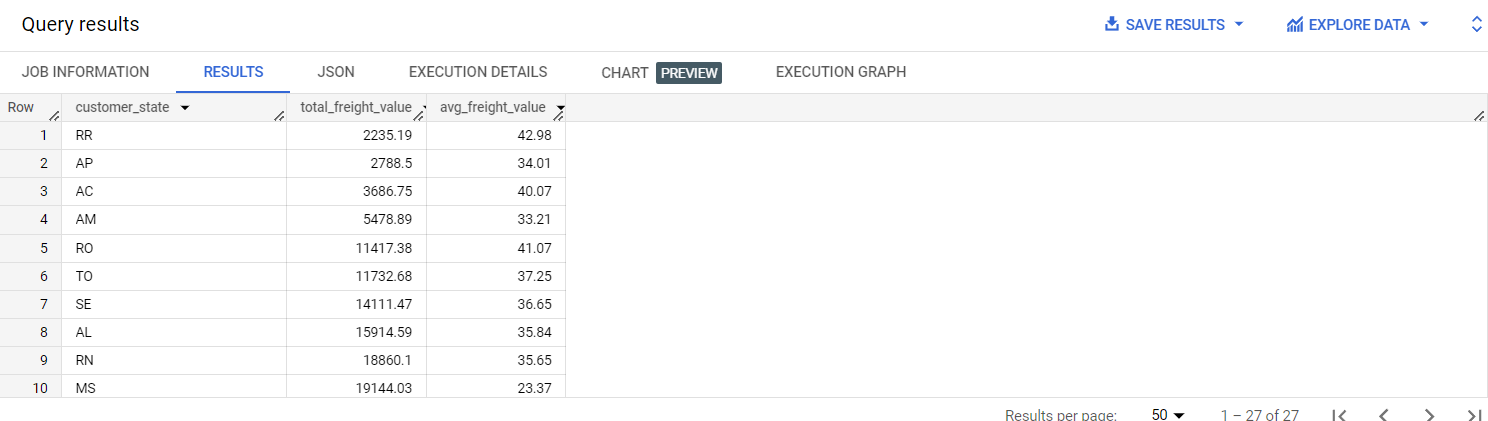
JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C.customer\_id

GROUP BY C.customer\_state

ORDER BY total\_freight\_value,avg\_freight\_value

Output:



**Insights** - highest total freight value is 19144.03 from MS state and highest average freight value 42.98 from state RR

**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.

Query:

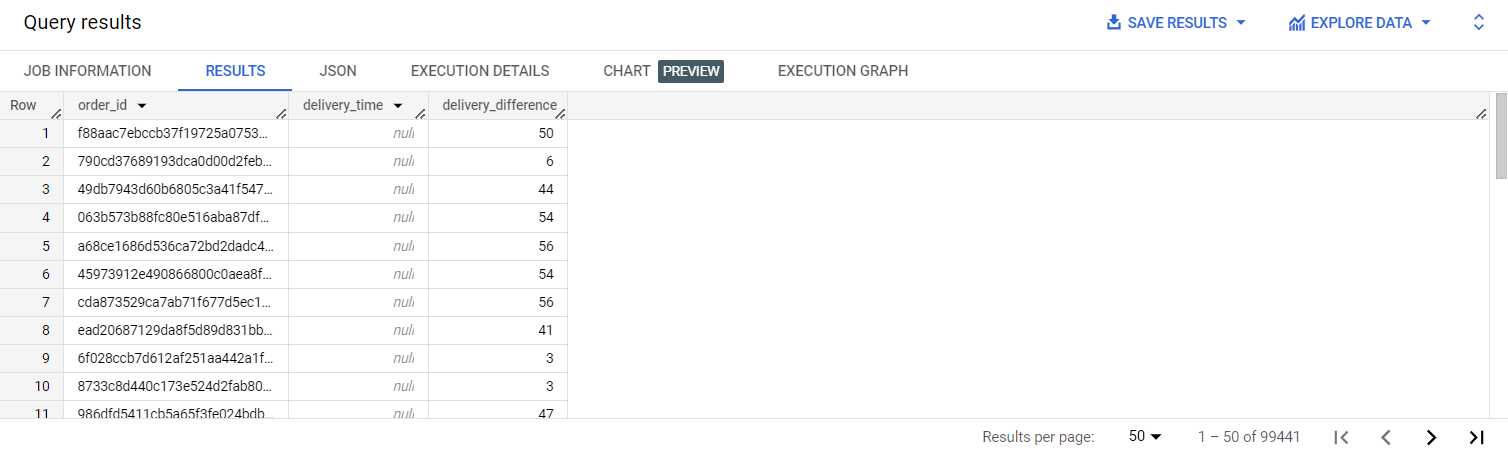
SELECT order\_id,

DATE\_DIFF(order\_purchase\_timestamp,order\_delivered\_customer\_date, DAY) AS delivery\_time,

DATE\_DIFF( order\_estimated\_delivery\_date,order\_purchase\_timestamp, DAY) AS delivery\_difference

FROM `ecommerce-399716.target.orders`

Output



**Insights**- least delivery difference is 3 and most is 56

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

Query

WITH average\_freight AS(

SELECT C.customer\_state,

AVG(OI.freight\_value) AS freightavg

FROM `ecommerce-399716.target.order\_items` AS OI

JOIN `ecommerce-399716.target.orders` AS O

ON OI.order\_id = O.order\_id

JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C.customer\_id

GROUP BY C.customer\_state

)

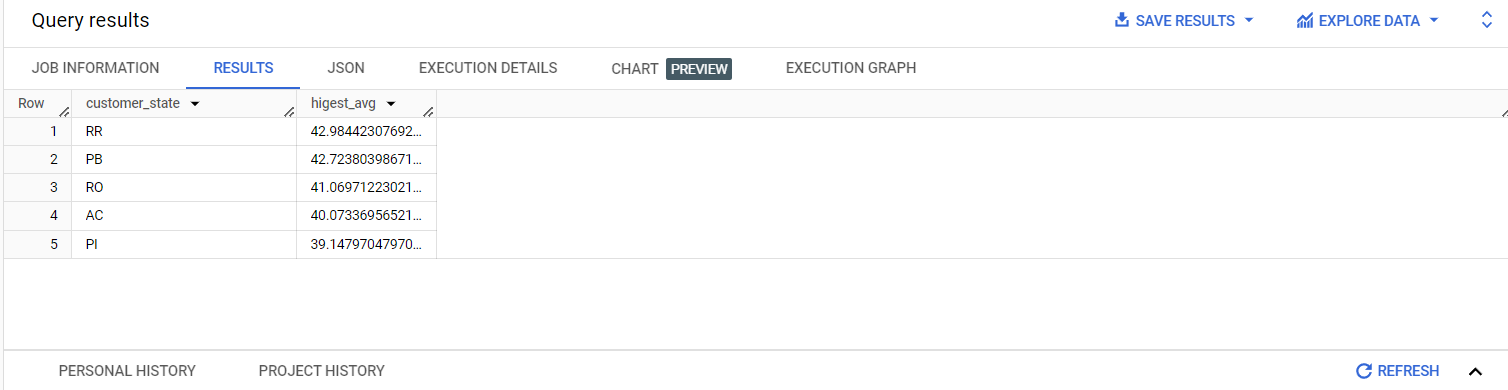
SELECT customer\_state,freightavg AS higest\_avg

FROM average\_freight

ORDER BY freightavg DESC

LIMIT 5

Output



WITH average\_freight AS(

SELECT C.customer\_state,

AVG(OI.freight\_value) AS freightavg

FROM `ecommerce-399716.target.order\_items` AS OI

JOIN `ecommerce-399716.target.orders` AS O

ON OI.order\_id = O.order\_id

JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C.customer\_id

GROUP BY C.customer\_state

)

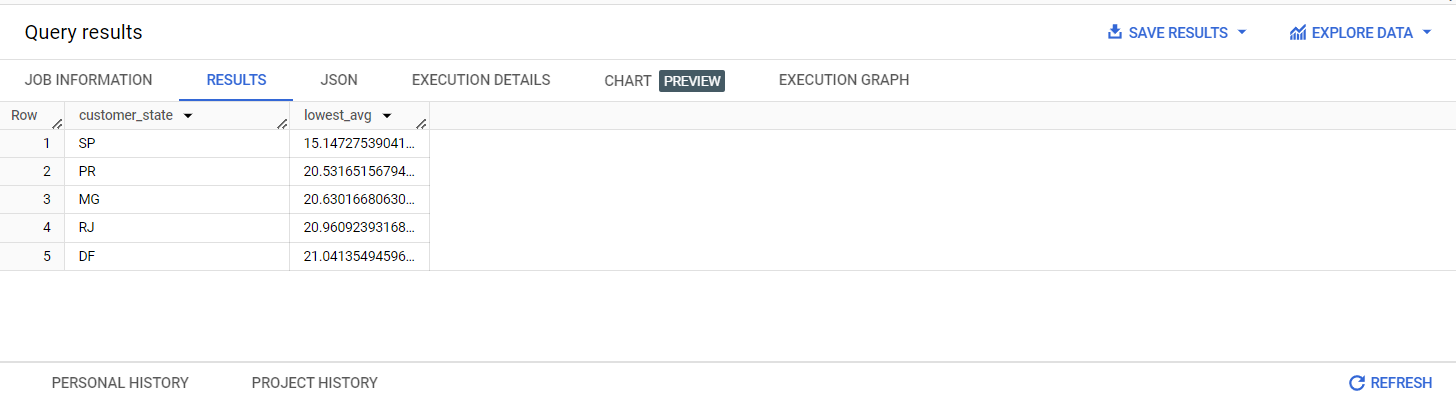
SELECT customer\_state,freightavg AS lowest\_avg

FROM average\_freight

ORDER BY freightavg ASC

LIMIT 5

Output



Note - I tried doing this with UNION ALL but it did not work so i had to writer 2 queries separately

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

Query

WITH avgdeliveytime AS (

SELECT C.customer\_state,

ROUND(AVG(DATE\_DIFF( O.order\_delivered\_customer\_date,O.order\_purchase\_timestamp, DAY))) AS highest\_avg\_delivery\_time

FROM `ecommerce-399716.target.orders` AS O

JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C.customer\_id

GROUP BY C.customer\_state

)

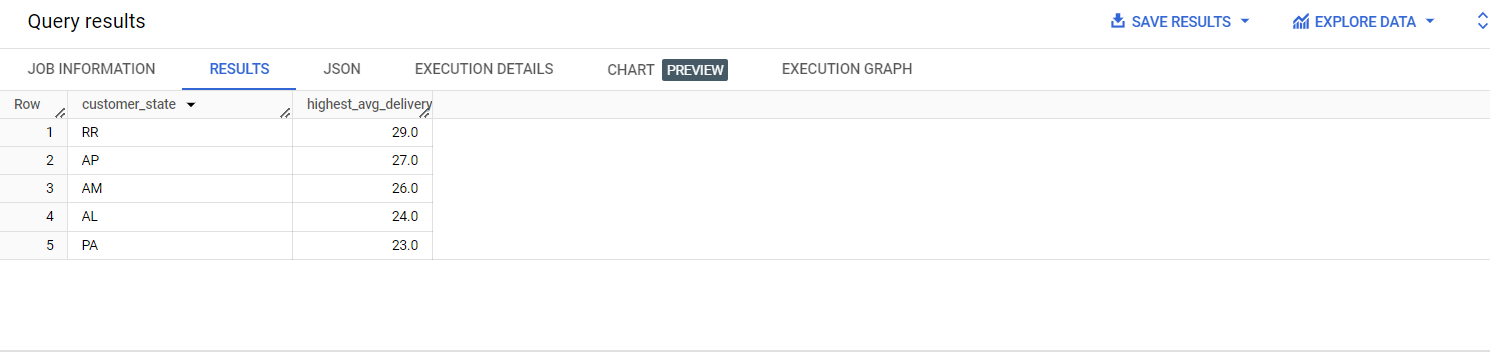
SELECT customer\_state,highest\_avg\_delivery\_time

FROM avgdeliveytime

ORDER BY highest\_avg\_delivery\_time DESC

LIMIT 5

Output



Query

WITH avgdeliveytime AS (

SELECT C.customer\_state,

ROUND(AVG(DATE\_DIFF( O.order\_delivered\_customer\_date,O.order\_purchase\_timestamp, DAY))) AS lowest\_avg\_delivery\_time

FROM `ecommerce-399716.target.orders` AS O

JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C.customer\_id

GROUP BY C.customer\_state

)

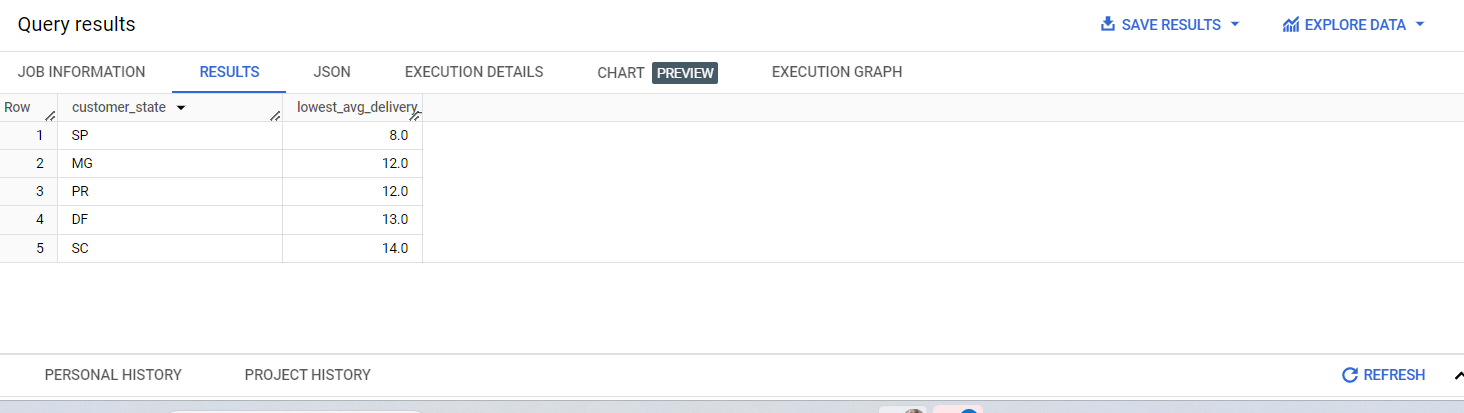
SELECT customer\_state,lowest\_avg\_delivery\_time

FROM avgdeliveytime

ORDER BY lowest\_avg\_delivery\_time ASC

LIMIT 5

Output



Note - I tried doing this with UNION ALL but it did not work so i had to writer 2 queries separately

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

WITH orderdelivery AS (

SELECT C.customer\_state,

ROUND(AVG(DATE\_DIFF(O.order\_estimated\_delivery\_date, O.order\_delivered\_customer\_date,DAY)),2) AS fastest\_delivery

FROM `ecommerce-399716.target.orders` AS O

JOIN `ecommerce-399716.target.customers` AS C

ON O.customer\_id = C. customer\_id

GROUP BY C.customer\_state

)

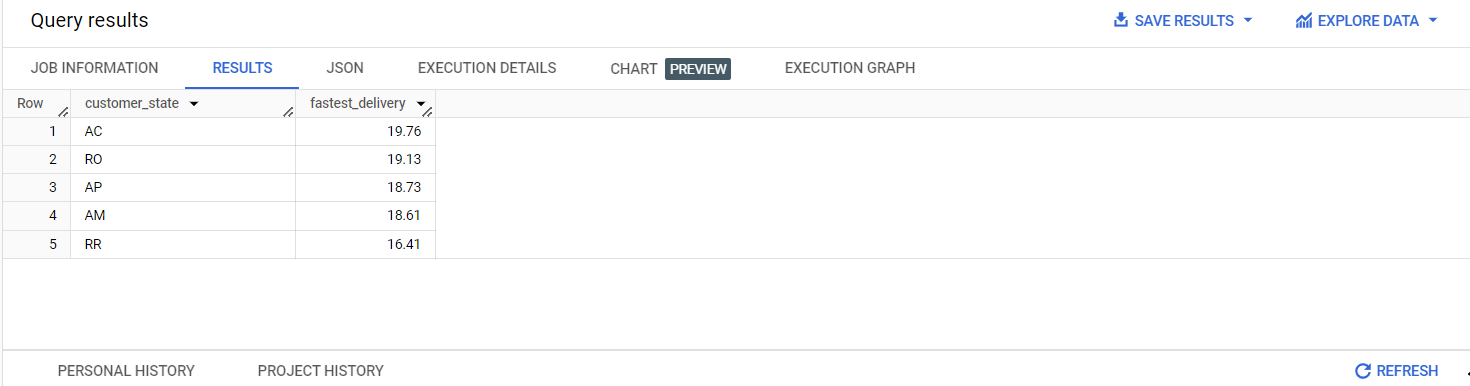
SELECT customer\_state, fastest\_delivery

FROM orderdelivery

ORDER BY fastest\_delivery DESC

LIMIT 5

Output



**Analysis based on the payments:**

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

Query

WITH MonthlyOrders AS (

SELECT

FORMAT\_DATE("%B", order\_purchase\_timestamp) AS order\_month,

payment\_type,

COUNT(\*) AS order\_count

FROM

`ecommerce-399716.target.orders` AS O

JOIN `ecommerce-399716.target.payments`AS P

ON O.order\_id = P.order\_id

GROUP BY

order\_month,

payment\_type

)

SELECT

order\_month,

payment\_type,

order\_count,

LAG(order\_count) OVER(PARTITION BY payment\_type ORDER BY order\_month) AS prev\_month\_order\_count

FROM

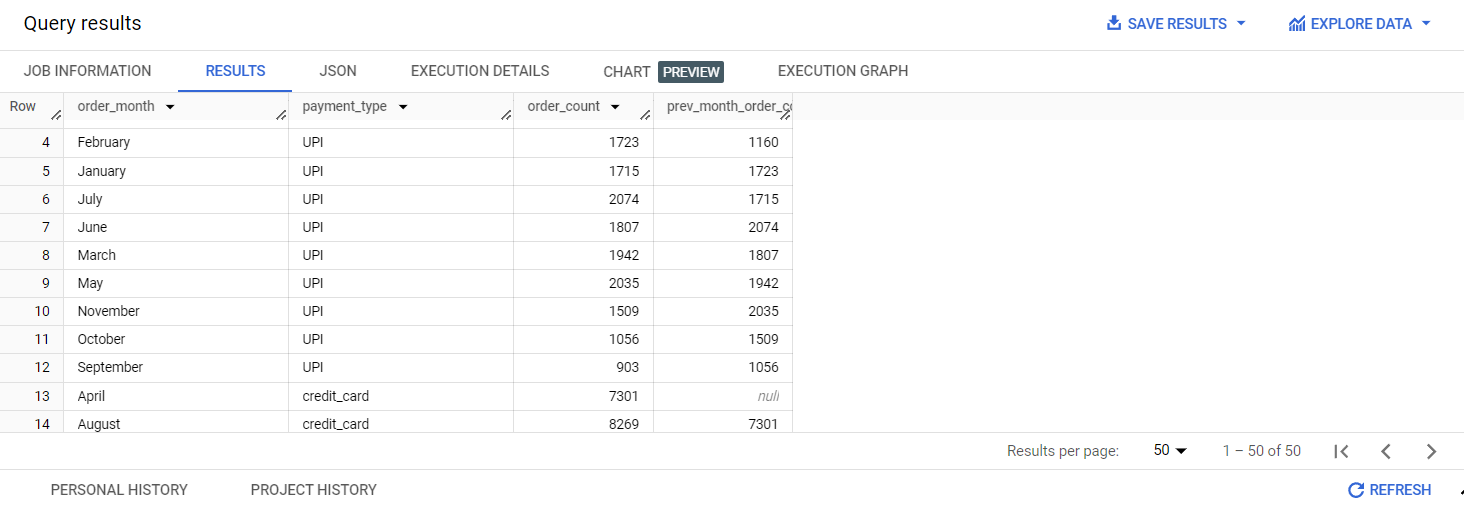
MonthlyOrders

ORDER BY

payment\_type,

Order\_month;

Output



**Insights** - mostly payment type used is UPI that is preferred payment mode

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

Query;

SELECT order\_id ,COUNT(DISTINCT payment\_sequential) AS paid\_installments

FROM `ecommerce-399716.target.payments`

GROUP BY order\_id

HAVING paid\_installments > 0

Output

