**Name: Karthick R**

**Project Platform: Big Query**

**PROJECT BIG BAZZAR**

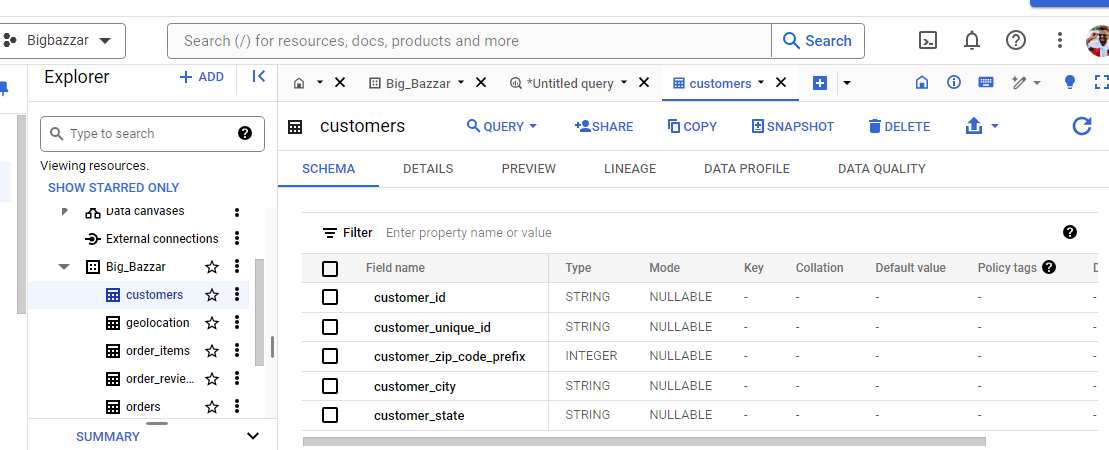
**Project Description:**

The Bigbazzar project examines Target's Brazilian operations through a dataset of 100,000 orders from 2016 to 2018. This analysis provides a detailed look at order status, pricing, payment and shipping performance, customer locations, product attributes, and reviews. By exploring this data, the project aims to uncover insights into order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and overall satisfaction in Target's Brazilian market.

Question:

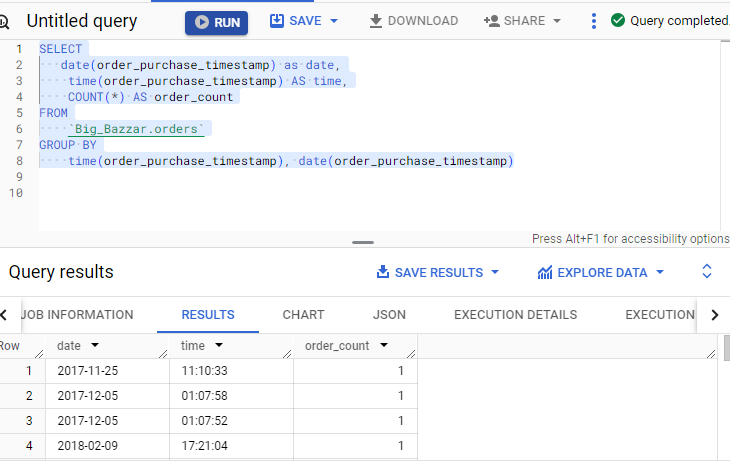
1. Data type of all columns in the "customers" table.

Output



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

Output



Query

SELECT

   date(order\_purchase\_timestamp) as date,

    time(order\_purchase\_timestamp) AS time,

    COUNT(\*) AS order\_count

FROM

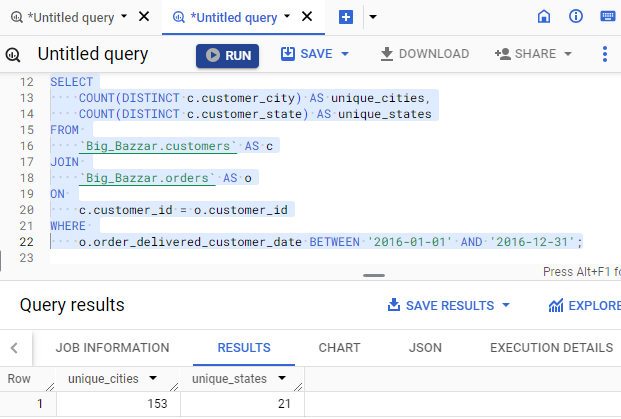
    `Big\_Bazzar.orders`

GROUP BY

    time(order\_purchase\_timestamp), date(order\_purchase\_timestamp);

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

Output



Query

SELECT

COUNT(DISTINCT c.customer\_city) AS unique\_cities,

COUNT(DISTINCT c.customer\_state) AS unique\_states

FROM

`Big\_Bazzar.customers` AS c

JOIN

`Big\_Bazzar.orders` AS o

ON

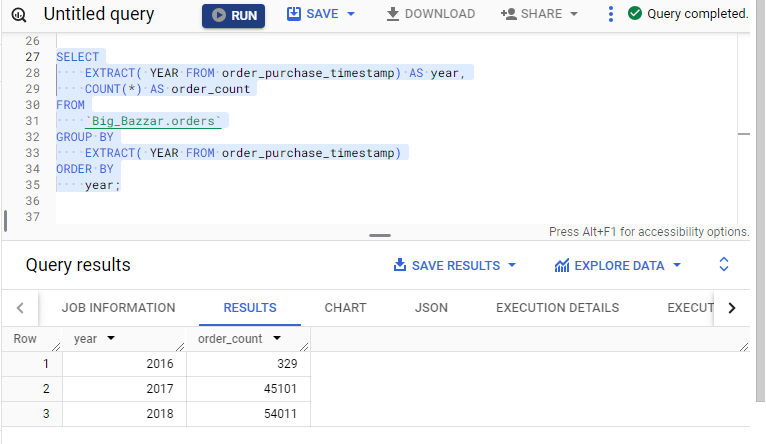
c.customer\_id = o.customer\_id

WHERE

o.order\_delivered\_customer\_date BETWEEN '2016-01-01' AND '2016-12-31';

**In-depth Exploration:**

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



SELECT

    EXTRACT( YEAR FROM order\_purchase\_timestamp) AS year,

    COUNT(\*) AS order\_count

FROM

    `Big\_Bazzar.orders`

GROUP BY

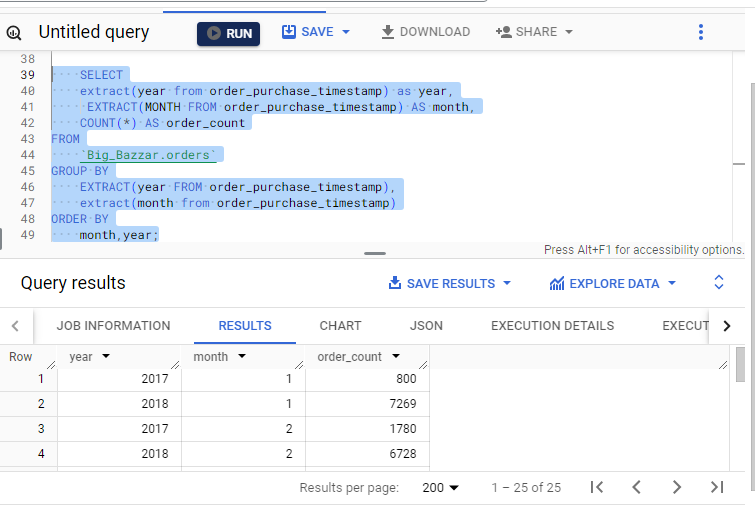
    EXTRACT( YEAR FROM order\_purchase\_timestamp)

ORDER BY

    year;

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

**OUTPUT**



**Query**

    SELECT

    extract(year from order\_purchase\_timestamp) as year,

     EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

    COUNT(\*) AS order\_count

FROM

    `Big\_Bazzar.orders`

GROUP BY

    EXTRACT(year FROM order\_purchase\_timestamp),

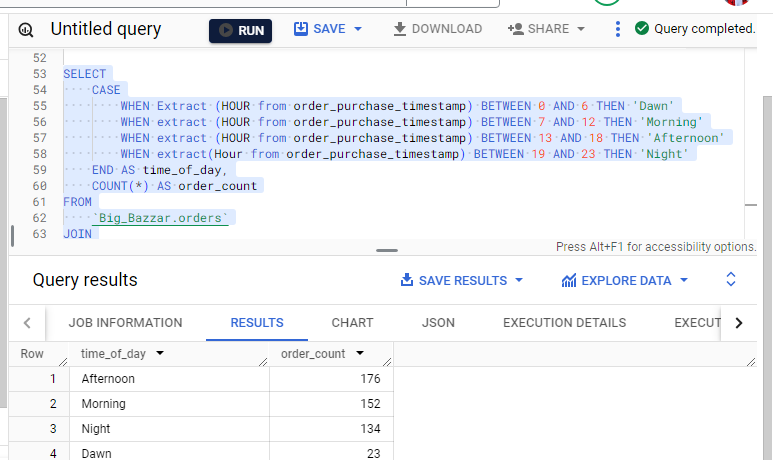
    extract(month from order\_purchase\_timestamp)

ORDER BY

    month,year;

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

Output



Query

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'

    END AS time\_of\_day,

    COUNT(\*) AS order\_count

FROM

    `Big\_Bazzar.orders`

JOIN

    `Big\_Bazzar.customers` ON `Big\_Bazzar.orders`.customer\_id = `Big\_Bazzar.customers`.customer\_id

WHERE

    customer\_state like 'RN'

GROUP BY

    time\_of\_day

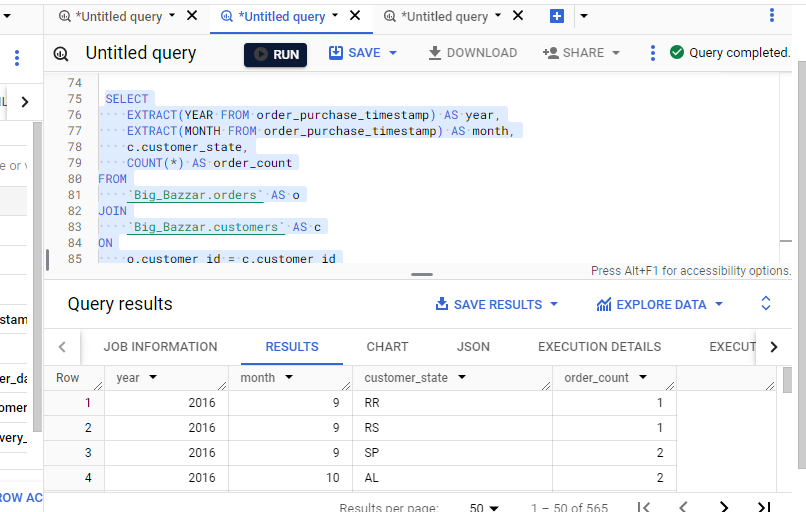
ORDER BY

    order\_count DESC;

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

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

**Output**



**Query**

SELECT

    EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year,

    EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

    c.customer\_state,

    COUNT(\*) AS order\_count

FROM

    `Big\_Bazzar.orders` AS o

JOIN

    `Big\_Bazzar.customers` AS c

ON

    o.customer\_id = c.customer\_id

GROUP BY

    year,

    month,

    customer\_state

ORDER BY

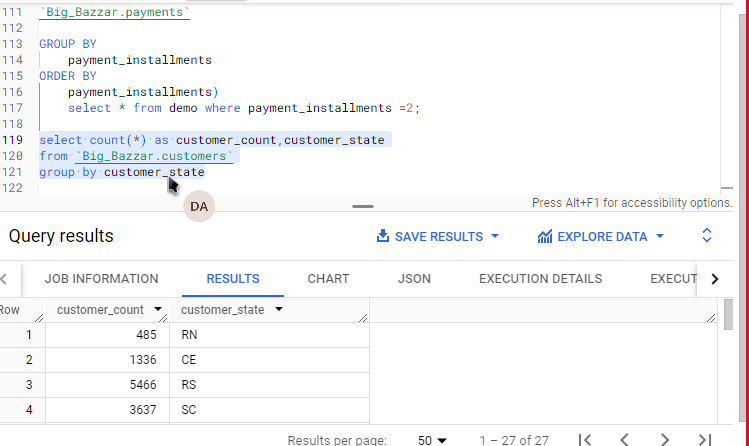
    year,

    month,

    customer\_state;

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

**Output**



**Query**

select count(\*) as customer\_count,customer\_state

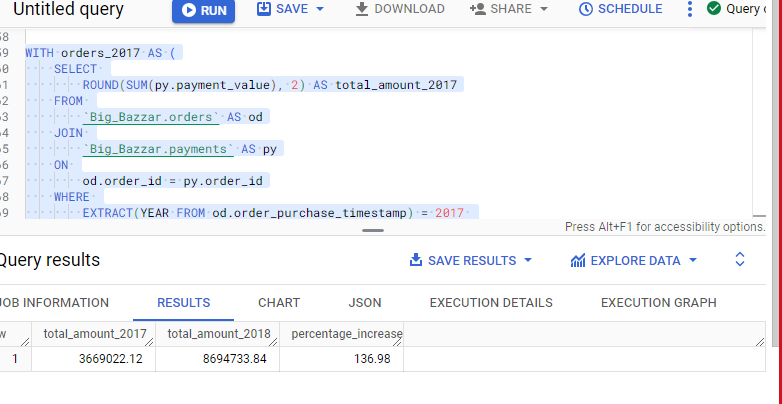
from `Big\_Bazzar.customers`

group by customer\_state

**4. 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).

**Output**



**Query**

WITH orders\_2017 AS (

SELECT

ROUND(SUM(py.payment\_value), 2) AS total\_amount\_2017

FROM

`Big\_Bazzar.orders` AS od

JOIN

`Big\_Bazzar.payments` AS py

ON

od.order\_id = py.order\_id

WHERE

EXTRACT(YEAR FROM od.order\_purchase\_timestamp) = 2017

AND EXTRACT(MONTH FROM od.order\_purchase\_timestamp) BETWEEN 1 AND 8

),

orders\_2018 AS (

SELECT

ROUND(SUM(py.payment\_value), 2) AS total\_amount\_2018

FROM

`Big\_Bazzar.orders` AS od

JOIN

`Big\_Bazzar.payments` AS py

ON

od.order\_id = py.order\_id

WHERE

EXTRACT(YEAR FROM od.order\_purchase\_timestamp) = 2018

AND EXTRACT(MONTH FROM od.order\_purchase\_timestamp) BETWEEN 1 AND 8

)

SELECT

o2017.total\_amount\_2017,

o2018.total\_amount\_2018,

ROUND(((o2018.total\_amount\_2018 - o2017.total\_amount\_2017) / o2017.total\_amount\_2017) \* 100, 2) AS percentage\_increase

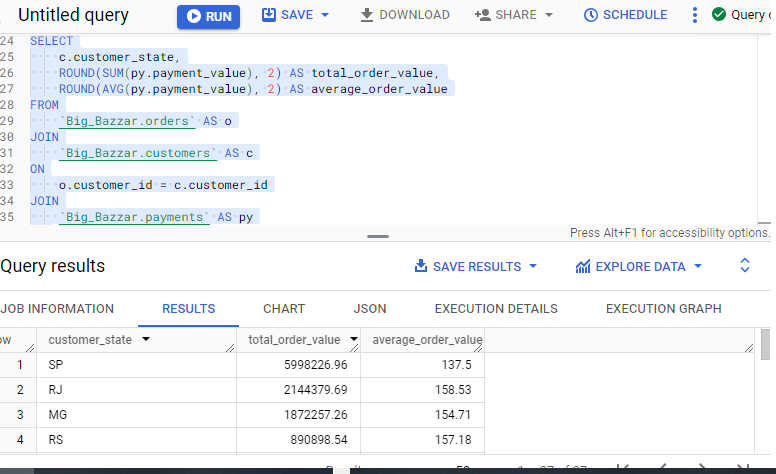
FROM

orders\_2017 AS o2017,

orders\_2018 AS o2018;

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

**Output**



**Query**

SELECT

c.customer\_state,

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

ROUND(AVG(py.payment\_value), 2) AS average\_order\_value

FROM

`Big\_Bazzar.orders` AS o

JOIN

`Big\_Bazzar.customers` AS c

ON

o.customer\_id = c.customer\_id

JOIN

`Big\_Bazzar.payments` AS py

ON

o.order\_id = py.order\_id

GROUP BY

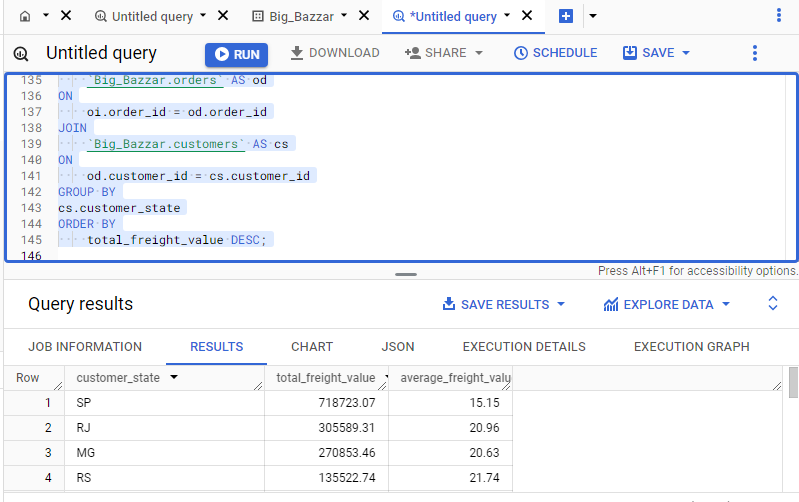
c.customer\_state

ORDER BY

total\_order\_value DESC;

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

**Output**



**Query**

SELECT

    cs.customer\_state,

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

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

FROM

    `Big\_Bazzar.order\_items` AS oi

JOIN

    `Big\_Bazzar.orders` AS od

ON

    oi.order\_id = od.order\_id

JOIN

    `Big\_Bazzar.customers` AS cs

ON

    od.customer\_id = cs.customer\_id

GROUP BY

cs.customer\_state

ORDER BY

    total\_freight\_value DESC;

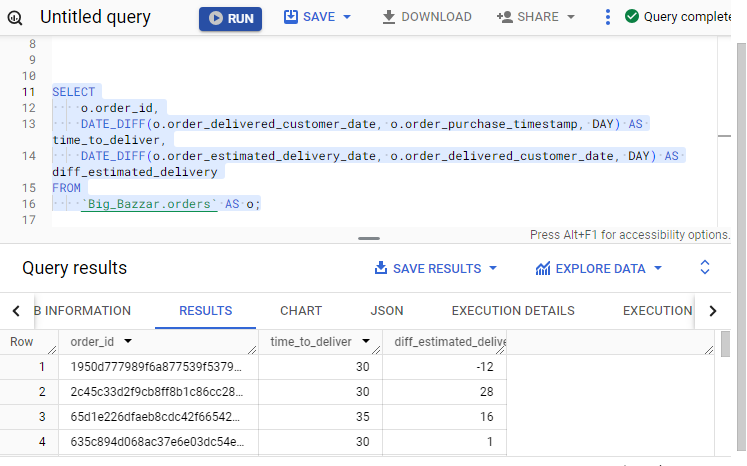
1. **Analysis based on sales, freight and delivery time.**

**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.  
  
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\_delivered\_customer\_date - order\_estimated\_delivery\_date

**Output**



**Query**

SELECT

    o.order\_id,

    DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY) AS time\_to\_deliver,

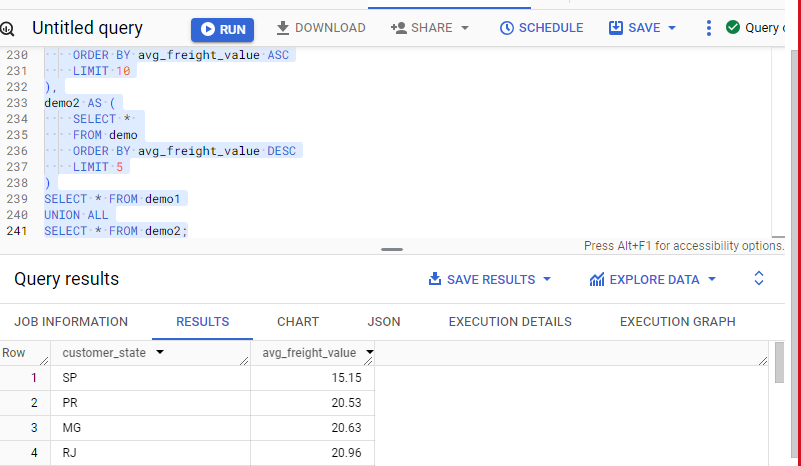
    DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date, DAY) AS diff\_estimated\_delivery

FROM

    `Big\_Bazzar.orders` AS o;

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

**Output**



**Query:**

WITH demo AS (

    SELECT

        c.customer\_state,

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

    FROM

        Big\_Bazzar.order\_items AS oi

    JOIN

        Big\_Bazzar.orders AS o ON oi.order\_id = o.order\_id

    JOIN

        Big\_Bazzar.customers AS c ON o.customer\_id = c.customer\_id

    GROUP BY

        c.customer\_state

),

demo1 AS (

    SELECT \*

    FROM demo

    ORDER BY avg\_freight\_value ASC

    LIMIT 10

),

demo2 AS (

    SELECT \*

    FROM demo

    ORDER BY avg\_freight\_value DESC

    LIMIT 5

)

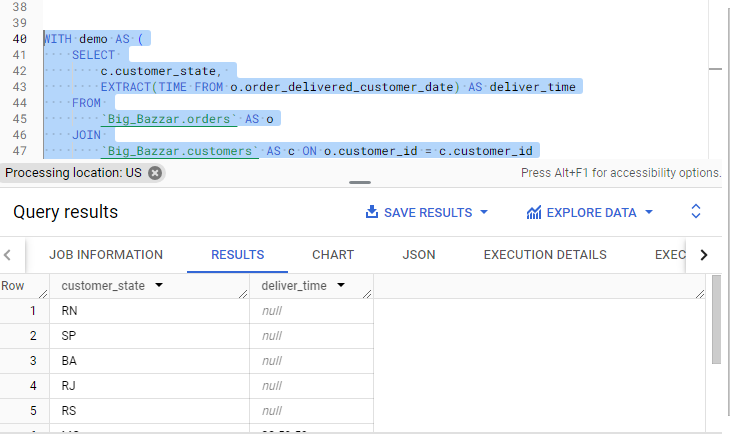
SELECT \* FROM demo1

UNION ALL

SELECT \* FROM demo2;

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

**Output**



**Query**

WITH demo AS (

    SELECT

        c.customer\_state,

        EXTRACT(TIME FROM o.order\_delivered\_customer\_date) AS deliver\_time

    FROM

        `Big\_Bazzar.orders` AS o

    JOIN

        `Big\_Bazzar.customers` AS c ON o.customer\_id = c.customer\_id

    GROUP BY

        c.customer\_state, deliver\_time

),

demo1 AS (

    SELECT

        customer\_state,

        deliver\_time

    FROM

        demo

    ORDER BY

        deliver\_time ASC

    LIMIT 5

),

demo2 AS (

    SELECT

        customer\_state,

        deliver\_time

    FROM

        demo

    ORDER BY

        deliver\_time DESC

    LIMIT 5

)

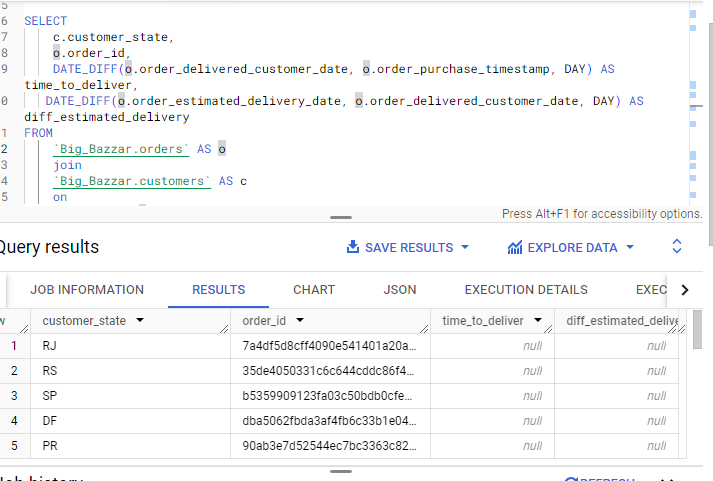
SELECT \* FROM demo1

UNION ALL

SELECT \* FROM demo2;

**D** .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.

**Output :**



**Query :**

SELECT

    c.customer\_state,

    o.order\_id,

    DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY) AS time\_to\_deliver,

   DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date, DAY) AS diff\_estimated\_delivery

FROM

    `Big\_Bazzar.orders` AS o

    join

    `Big\_Bazzar.customers` AS c

    on

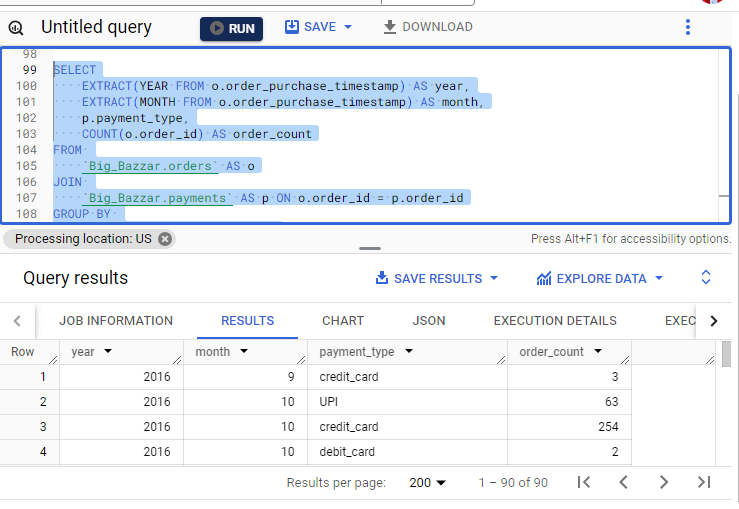
c.customer\_id = o.customer\_id

limit 5

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

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

**Output**



**Query**

SELECT

    EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS year,

    EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,

    p.payment\_type,

    COUNT(o.order\_id) AS order\_count

FROM

    `Big\_Bazzar.orders` AS o

JOIN

    `Big\_Bazzar.payments` AS p ON o.order\_id = p.order\_id

GROUP BY

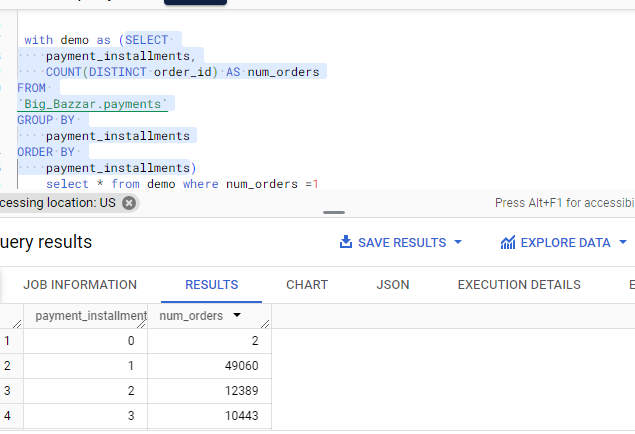
    year, month, p.payment\_type

ORDER BY

    year, month, p.payment\_type;

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

**Output**



**Query**

with demo as (SELECT

    payment\_installments,

    COUNT(DISTINCT order\_id) AS num\_orders

FROM

`Big\_Bazzar.payments`

GROUP BY

    payment\_installments

ORDER BY

    payment\_installments)

    select \* from demo where num\_orders =1