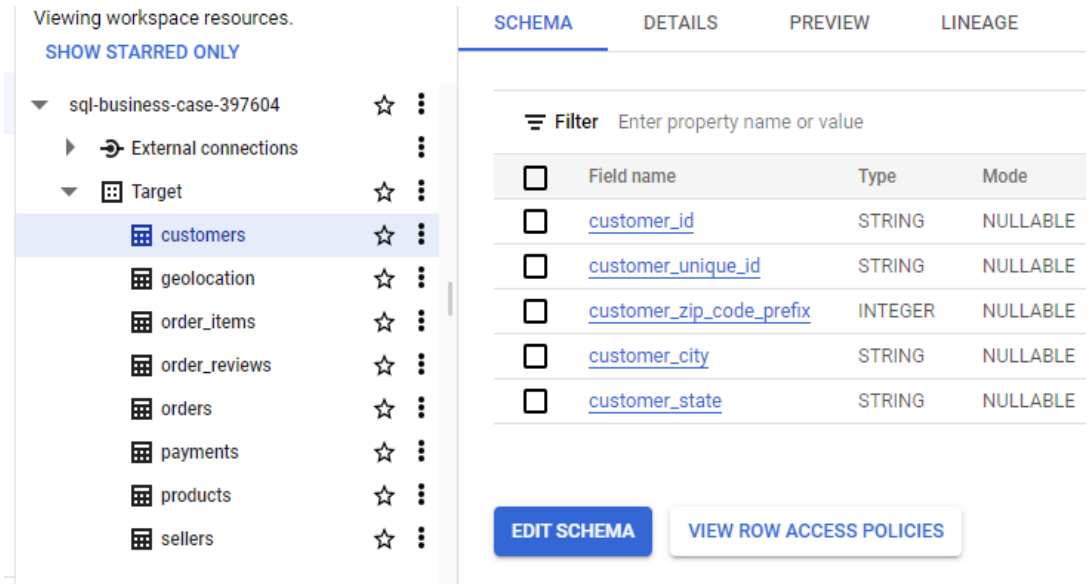
TARGET – BUSINESS CASE – SQL

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

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

OUTPUT:



INFERENCE:

- The above image shows that the **Data type** of **all columns** in the **customers** table.
- In that **customers** table contains 2 types of Data types.
  (i.e)
  1. **STRING** - (It stores the **Variables**)
  2. **INTEGER** - (It contains only **Numbers**)

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

QUERY:

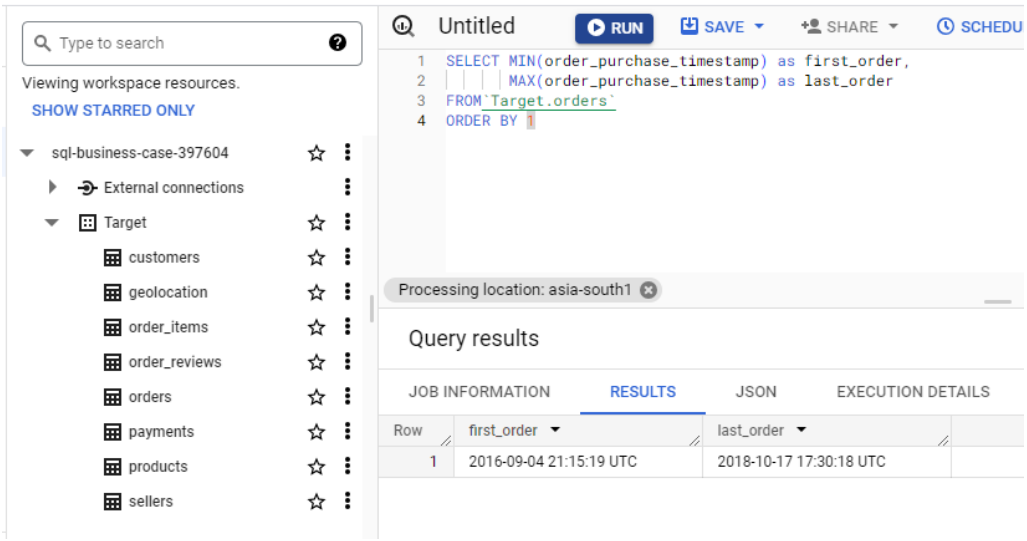
```
SELECT MIN(order_purchase_timestamp) as first_order,

MAX(order_purchase_timestamp) as last_order

FROM`Target.orders`

ORDER BY 1;
```

OUTPUT:



INFERENCE:

- The Above image shows the result of Time Range between the First order and Last order.
    - **First order** date on **04-09- 2016** (September 4th 2016) and time on **21:15:19 UTC**
    - **Last order** date on **17-10-2018** (October 17th 2018) and time on **17:30:18 UTC**
- The Total Duration between the **First order and Last order** is **2 Years and 1 Month**.

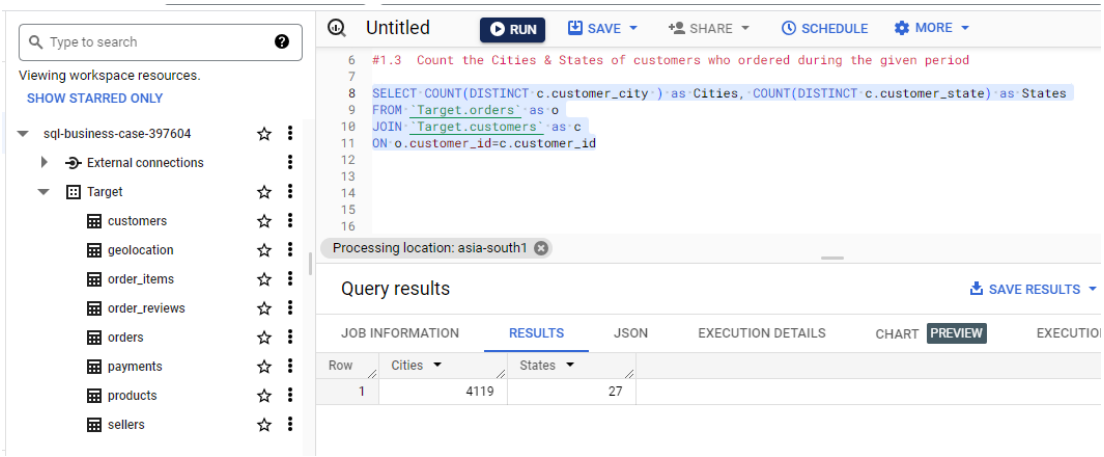
1.3 Count the Cities & States of customers who ordered during the given period.

QUERY:

```
SELECT COUNT(DISTINCT c.customer_city ) as Cities, COUNT(DISTINCT c.customer_state) as States

FROM `Target.orders` as o
JOIN `Target.customers` as c
ON o.customer_id=c.customer_id
```

OUTPUT:



INFERENCE:

- The Above query finds the **Total Number** of **Cities** and **States** of **Customers purchased** in the **Given Period.**
    - **4119 Cities**
    - **27 States**

2. In-depth Exploration

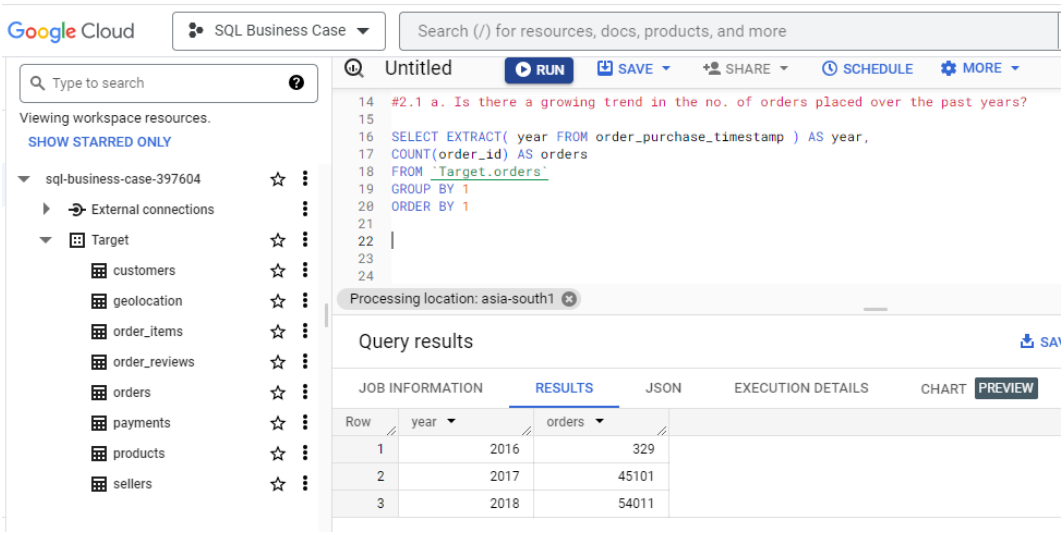
2.1 Is there a growing trend in the no. of orders placed over the past years?

QUERY:

```
SELECT EXTRACT( year FROM order_purchase_timestamp ) AS year,

COUNT(order_id) AS orders
FROM `Target.orders`
GROUP BY 1
ORDER BY 1
```

OUTPUT:



INFERENCE:

- After the analysis of Dataset,
  - During the **First Year 2016**, the total number of orders was **329**.
  - But in the **Second Year 2017** it has an increase by **45101** total number of orders.
  - Then we have **54011** total number of orders in the **Last Year 2018** which is the **highest number of orders among the three years**.
- Finally we can clearly understand that the **number of orders** has a **growing trend** over the given period of time.

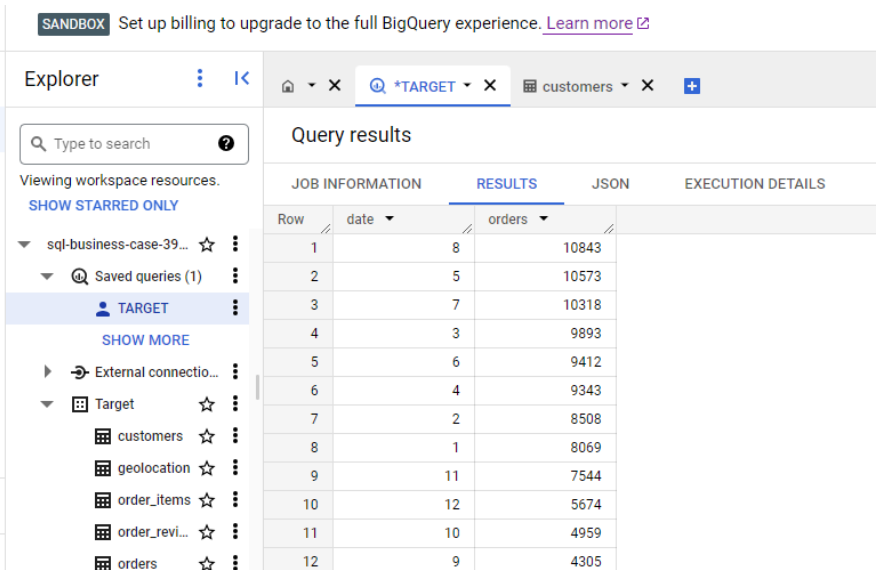
2.2Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

QUERY:

```
SELECT EXTRACT(month FROM order_purchase_timestamp) AS date,

COUNT(order_id) AS orders
FROM `Target.orders`
GROUP BY date
ORDER BY orders DESC
```

OUTPUT:



Row	date	orders
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674
11	10	4959
12	9	4305

INFERENCE:

- The Above Image shows the number of orders placed in monthly seasonality. Here the Top 3 months of orders,
  - During **August** the highest number of orders were placed – **10843 orders.**
  - The **Second highest** orders placed is on **May** – **10573 orders.**
  - Third highest number of orders were placed on **July – 10318 orders.**

2.3During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

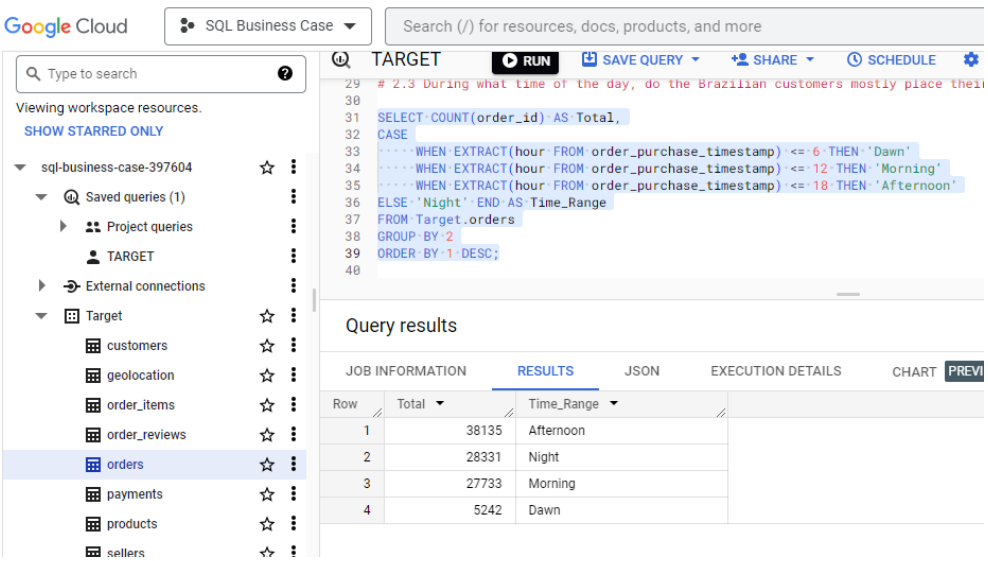
- **0-6 hrs : Dawn**
- **7-12 hrs : Mornings**
- **13-18 hrs : Afternoon**
- **19-23 hrs : Night**

QUERY:

```
SELECT COUNT(order_id) AS Total,

CASE
      WHEN EXTRACT(hour FROM order_purchase_timestamp) <= 6 THEN 'Dawn'
      WHEN EXTRACT(hour FROM order_purchase_timestamp) <= 12 THEN 'Morning'
      WHEN EXTRACT(hour FROM order_purchase_timestamp) <= 18 THEN 'Afternoon'
      ELSE 'Night' END AS Time_Range
FROM Target.orders
GROUP BY 2
ORDER BY 1 DESC;
```

OUTPUT:



INFERENCE:

- After the analysis of dataset, mostly **Brazilian Customers** done the orders in **Peak Time** of **Afternoon** compared to the other timings**.**
  - **Afternoon – 38135** orders
  - **Night – 28331** orders
  - **Morning – 27733** orders
  - **Dawn – 5242** orders

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

3.1Get the month on month no. of orders placed in each state.

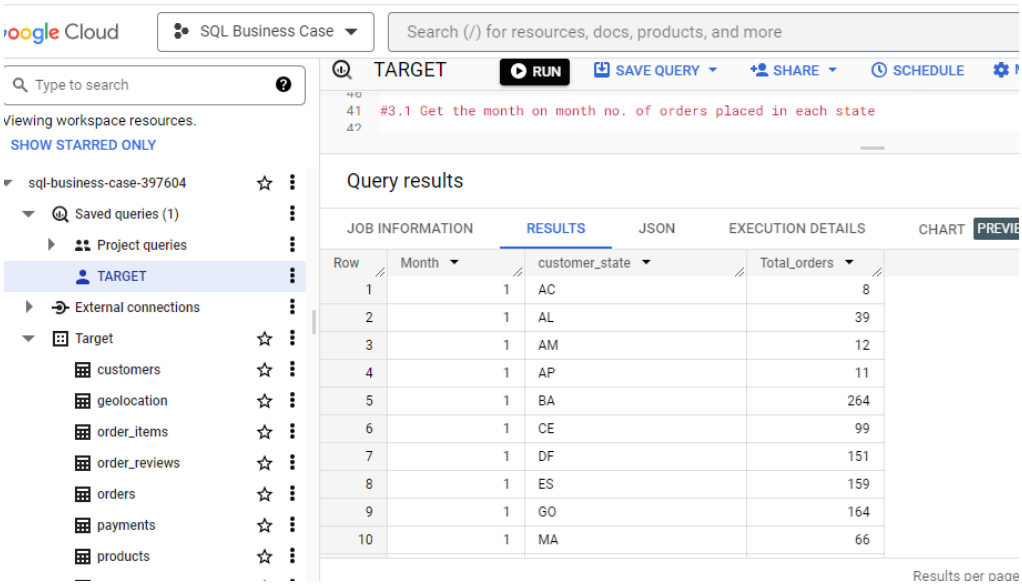
QUERY:

```
SELECT EXTRACT(month FROM order_purchase_timestamp) AS Month, C.customer_state,

       COUNT(order_id) AS Total_orders

FROM `Target.customers` C
JOIN `Target.orders` O
ON C.customer_id = O.customer_id
GROUP BY 1, 2
ORDER BY 1,2
```

OUTPUT:



INFERENCE:

- After analyzing the dataset the **Total Number of orders** placed in **each state on month on month** in the above Image.
    - No. of orders are grouped by **Month on Month** in each state.

3.2How are the customers distributed across all the states?

QUERY:

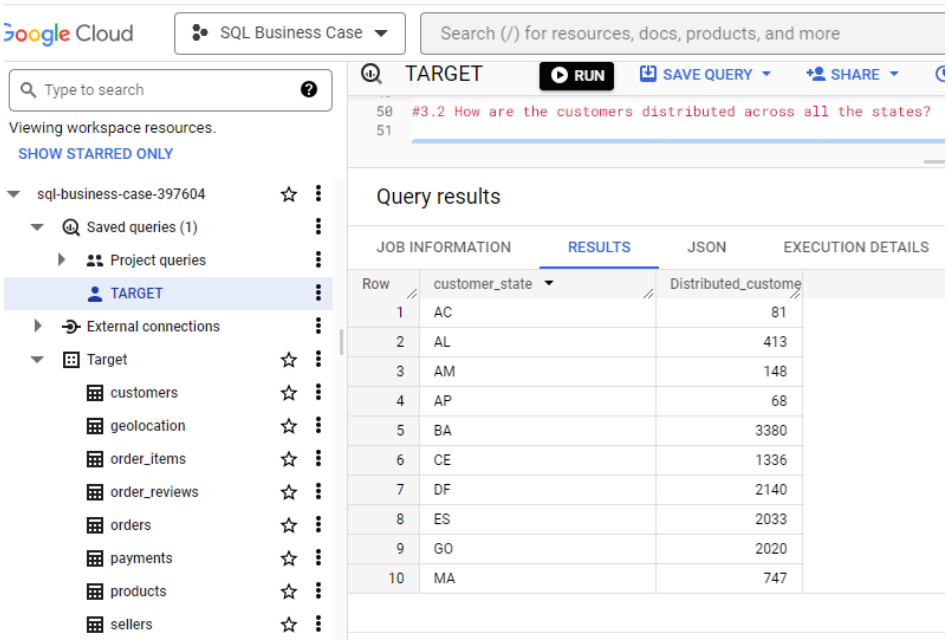
```
SELECT customer_state, COUNT(DISTINCT customer_id) AS Distributed_customers

FROM `Target.customers`

GROUP BY 1

ORDER BY 1
```

OUTPUT:



INFERENCE:

- After the analysis of dataset, the Distributed Customer present in **all states** is displayed through the above query and the output is displayed in the above image.
    - The **distributed customers** are grouped by the **unique customer in each state.**

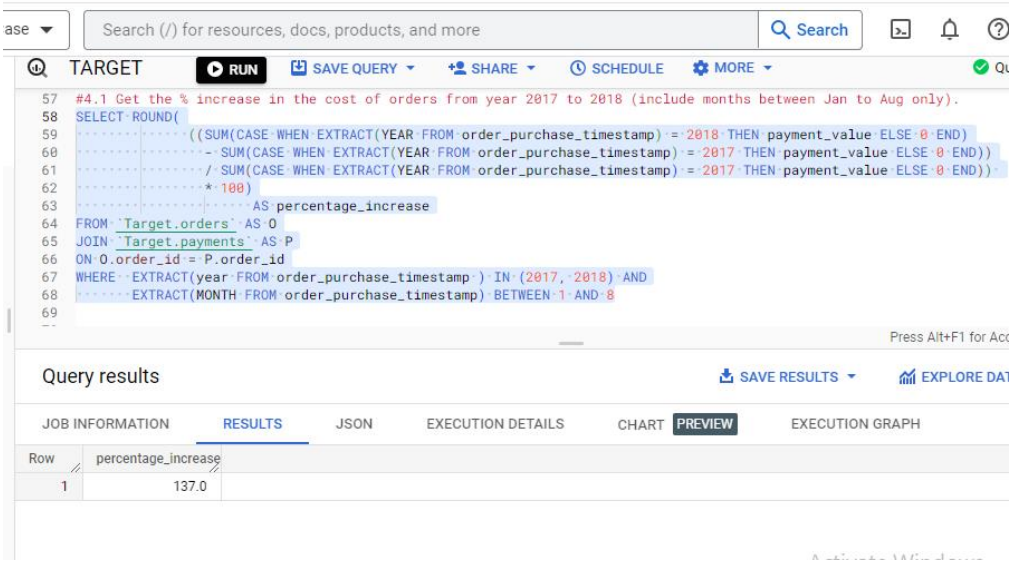
4. Impact on Economy: Analyze the money movement by Ecommerce by looking at order prices, freight and others.

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

QUERY:

```
SELECT ROUND(
        ((SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2018 THEN payment_value ELSE 0
END)
       - SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 THEN payment_value ELSE 0
END))
       / SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 THEN payment_value ELSE 0
END))
       * 100)
            AS percentage_increase
FROM `Target.orders` AS O
JOIN `Target.payments` AS P
ON O.order_id = P.order_id
WHERE EXTRACT(year FROM order_purchase_timestamp ) IN (2017, 2018) AND
      EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8
```

OUTPUT:



INFERENCE:

- After the analysis of the dataset, the **Percentage Increase** in the **cost of orders** from the **year 2017 to 2018** (includes the **months** between **Jan to Aug only**) is **137%.**

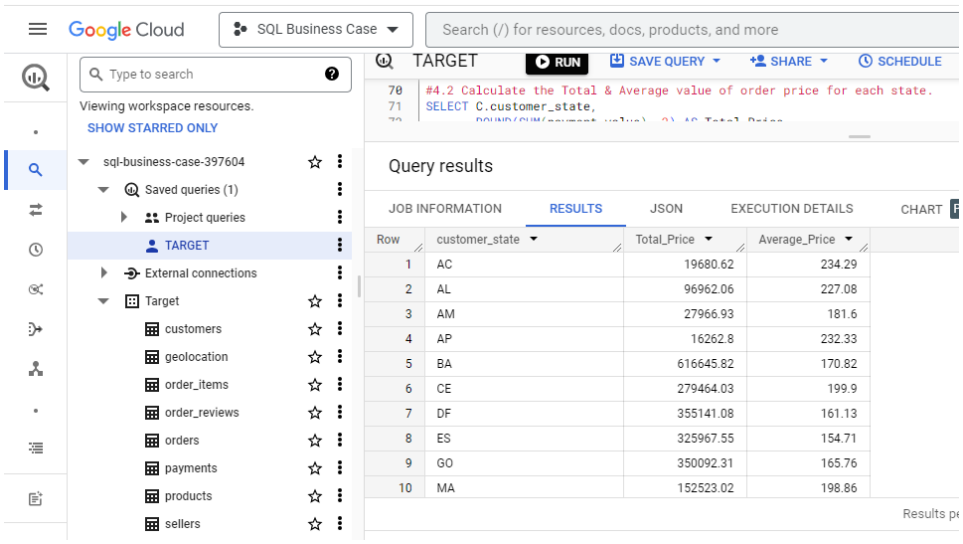
4.2Calculate the Total & Average value of order price for each state.

QUERY:

```
SELECT C.customer_state,

       ROUND(SUM(payment_value), 2) AS Total_Price,
       ROUND(AVG(payment_value), 2) AS Average_Price
FROM `Target.customers` C
JOIN `Target.orders`O
ON C.customer_id = O.customer_id
JOIN `Target.payments` P
ON P.order_id = O.order_id
GROUP BY 1
ORDER BY 1
```

OUTPUT:



INFERENCE:

- After the analysis of the dataset, the **Total and Average** value of **order price** for each state as shown in the output image.
    - The **Total Price** and the **Average Price** of the **orders** are grouped by **each State** respectively.

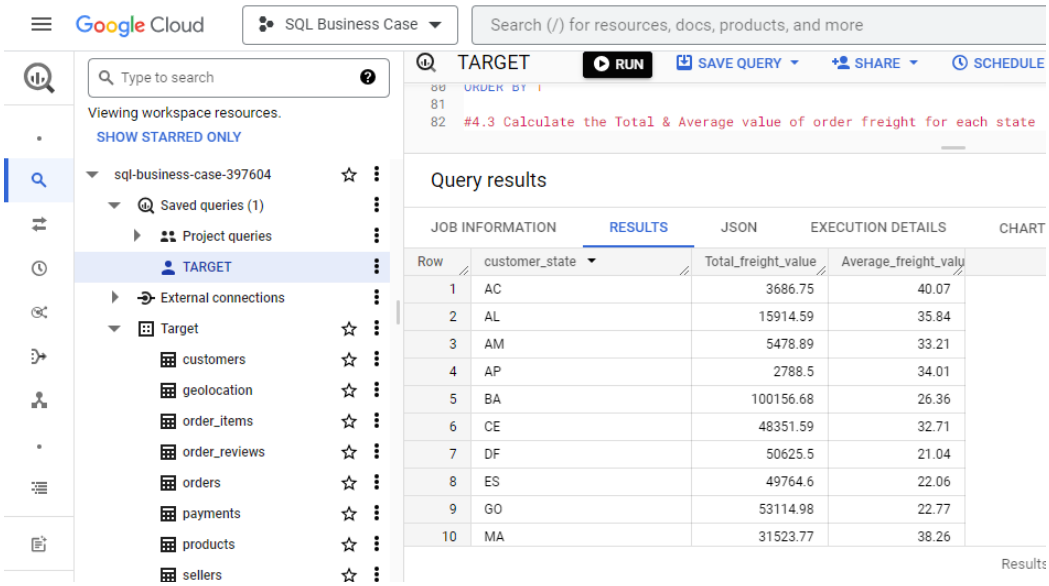
4.3 Calculate the Total & Average value of order freight for each state.

QUERY:

```
SELECT C.customer_state,

       ROUND(SUM(freight_value), 2) AS Total_freight_value,
       ROUND(AVG(freight_value), 2) AS Average_freight_value
FROM `Target.customers` C
JOIN `Target.orders`O
ON C.customer_id = O.customer_id
JOIN `Target.order_items` OI
ON OI.order_id = O.order_id
GROUP BY 1
ORDER BY 1
```

OUTPUT:



INFERENCE:

- After the analysis of the dataset, the **Total and Average** value of **order freight** for each state as shown in the output image.
    - The **Total freight value** and the **Average freight value** of the **orders** are grouped by **each State** respectively.

5. Analysis based on sales, freight and delivery time

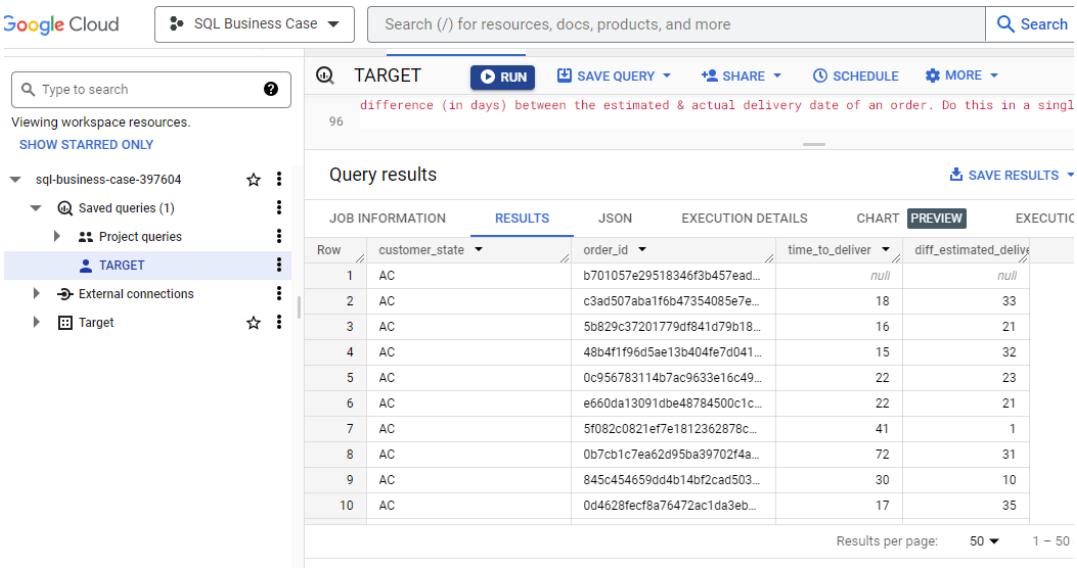
5.1 Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

QUERY:

```
        SELECT customer_state,

               order_id,
               DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) AS
time_to_deliver,
             ABS(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day))
AS diff_estimated_delivery
        FROM `Target.orders` AS O
        JOIN `Target.customers` AS C
        ON O.customer_id = C.customer_id
        GROUP BY 1,2,3,4
        ORDER BY 1
```

OUTPUT:



INFERENCE:

- The no. of days taken to deliver each order from the order's purchase date is shown as **time_to_deliver**.
- And the calculated difference between the estimated & actual delivery date of an order is shown as **diff_estimated_delivey**.

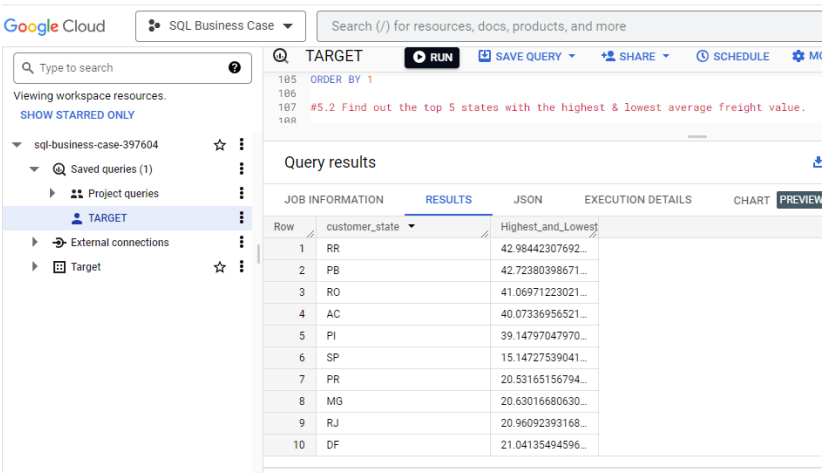
5.2 Find out the top 5 states with the highest & lowest average freight value.

QUERY:

```
        (SELECT customer_state, AVG(OI.freight_value) AS Highest_and_Lowest_freight_value

         FROM `Target.orders` O
         JOIN `Target.customers` C
         ON O.customer_id = C.customer_id
         JOIN `Target.order_items` OI
         ON O.order_id = OI.order_id
         GROUP BY 1
         ORDER BY 2 DESC
         LIMIT 5)
UNION DISTINCT
        (SELECT customer_state, AVG(OI.freight_value) AS Highest_and_Lowest_freight_value
         FROM `Target.orders` O
         JOIN `Target.customers` C
         ON O.customer_id = C.customer_id
         JOIN `Target.order_items` OI
         ON O.order_id = OI.order_id
         GROUP BY 1
         ORDER BY 2 ASC
         LIMIT 5)
```

OUTPUT:



INFERENCE:

- After the analysis of dataset, The **Top 5 States** with the **Highest and Lowest Average freight value** is displayed in the above output,
    - The **First 5** output values from **1 to 5** shows the **Highest Average freight values.**
    - The **Last 5** output values from **6-10** shows the **Lowest Average freight values**.

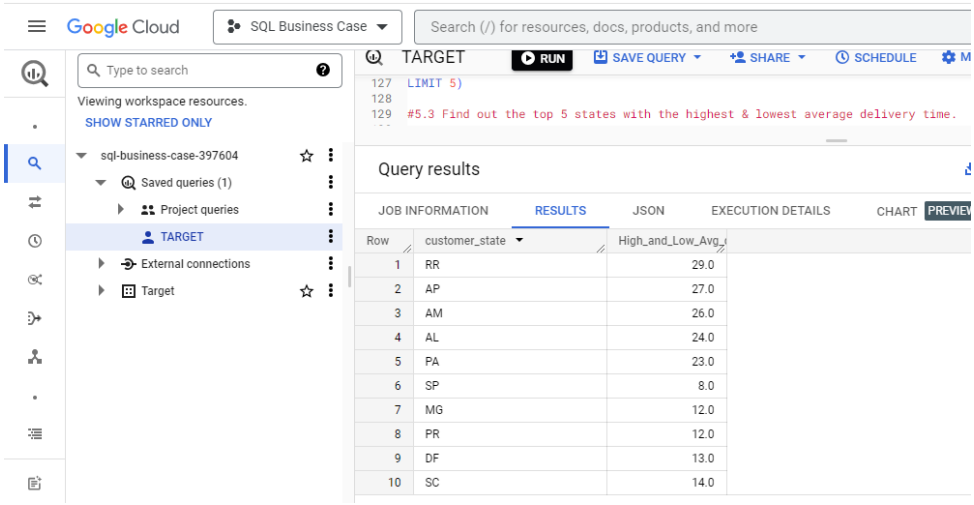
5.3 Find out the top 5 states with the Highest & Lowest average delivery time.

QUERY:

```
            (SELECT C.customer_state, ROUND(AVG(time_to_deliver)) AS High_and_Low_Avg_delivery_time

             FROM (SELECT *, DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) AS
time_to_deliver
                   FROM `Target.orders`) O
             JOIN `Target.customers` C
             ON C.customer_id = O.customer_id
             GROUP BY 1
             ORDER BY 2 DESC
             LIMIT 5)
   UNION ALL
            (SELECT C.customer_state, ROUND(AVG(time_to_deliver)) AS High_and_Low_Avg_delivery_time
             FROM (SELECT *, DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) AS
time_to_deliver
                   FROM `Target.orders`) O
            JOIN `Target.customers` C
            ON C.customer_id = O.customer_id
            GROUP BY 1
            ORDER BY 2
            LIMIT 5)
```

OUTPUT:



Row	customer_state	High_and_Low_Avg_
1	RR	29.0
2	AP	27.0
3	AM	26.0
4	AL	24.0
5	PA	23.0
6	SP	8.0
7	MG	12.0
8	PR	12.0
9	DF	13.0
10	SC	14.0

INFERENCE:

- After the analysis of dataset, The **Top 5 States** with the **Highest and Lowest Average delivery time** is displayed in the above output,
  - The **First 5** output values from **1 to 5** shows the **Highest Average delivery time (In Days).**
  - The **Last 5** output values from **6-10** shows the **Lowest Average delivery time (In Days).**

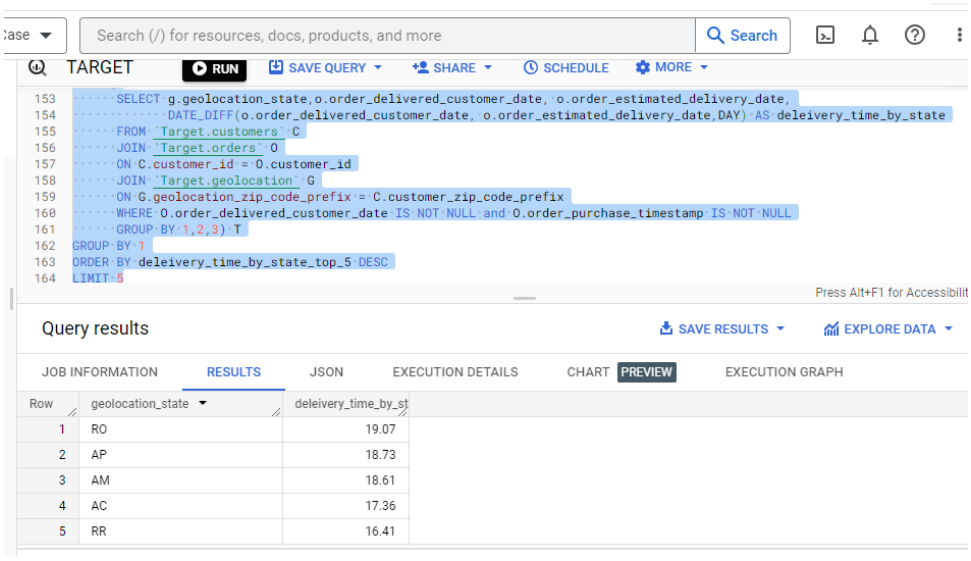
5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

QUERY:

```
            SELECT geolocation_state, ROUND(ABS(AVG(deleivery_time_by_state)),2) AS
deleivery_time_by_state_top_5

            FROM (
                  SELECT g.geolocation_state,o.order_delivered_customer_date, o.order_estimated_delivery_date,
                        DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date,DAY) AS
deleivery_time_by_state
                  FROM `Target.customers` C
                  JOIN `Target.orders` O
                  ON C.customer_id = O.customer_id
                  JOIN `Target.geolocation` G
                  ON G.geolocation_zip_code_prefix = C.customer_zip_code_prefix
                  WHERE O.order_delivered_customer_date IS NOT NULL and O.order_purchase_timestamp IS NOT NULL
                  GROUP BY 1,2,3) T
            GROUP BY 1
            ORDER BY deleivery_time_by_state_top_5 DESC
            LIMIT 5
```

OUTPUT:



Row	geolocation_state	deleivery_time_by_st
1	RO	19.07
2	AP	18.73
3	AM	18.61
4	AC	17.36
5	RR	16.41

INFERENCE:

- After the analysis o dataset, the **Top 5 States** where the **order delivery** is **really fast** as compared to the estimated date of delivery are shown in the above output image.

6. Analysis based on the payments

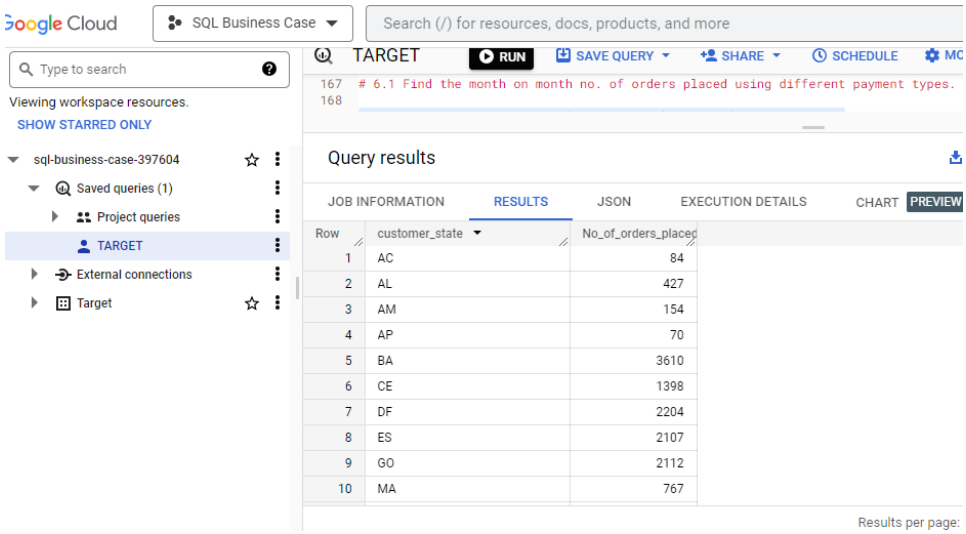
6.1 Find the month on month no. of orders placed using different payment types.

QUERY:

```
SELECT customer_state, COUNT(payment_type) AS No_of_orders_placed

FROM `Target.customers` C
JOIN `Target.orders` O
ON C.customer_id = O.customer_id
JOIN `Target.payments` P
ON P.order_id = O.order_id
GROUP BY 1
ORDER BY 1
```

OUTPUT:



INFERENCE:

- After analyzing the dataset, the **Number of orders** placed using **different payment methods** in **each state on month on month** in the above Image.
    - No. of orders are grouped by **Month on Month** in each state.

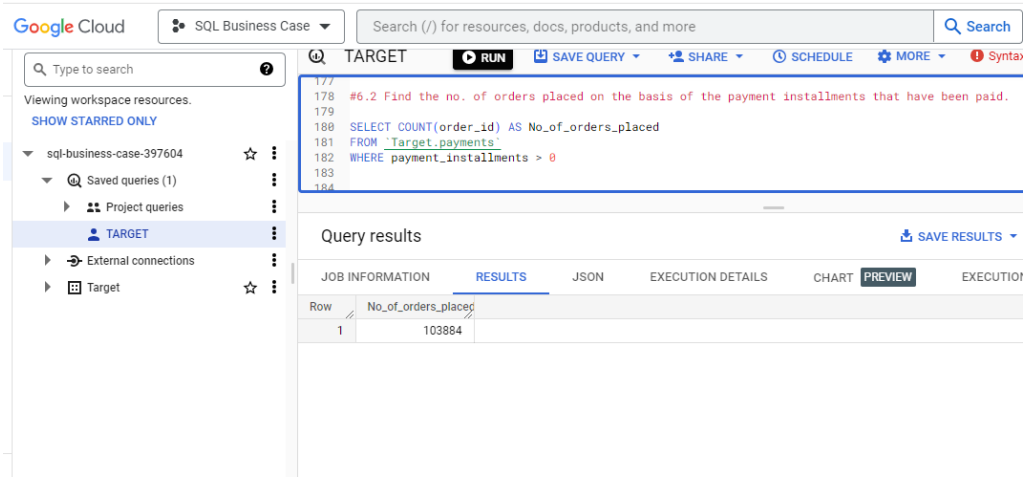
6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

QUERY:

```
SELECT COUNT(order_id) AS No_of_orders_placed

FROM `Target.payments`
WHERE payment_installments > 0
```

OUTPUT:



INFERENCE:

- After the analysis of dataset, the **Total Number of orders** placed on the basis of the **payment installments** that have paid is **103884.**