Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

Dear Team,

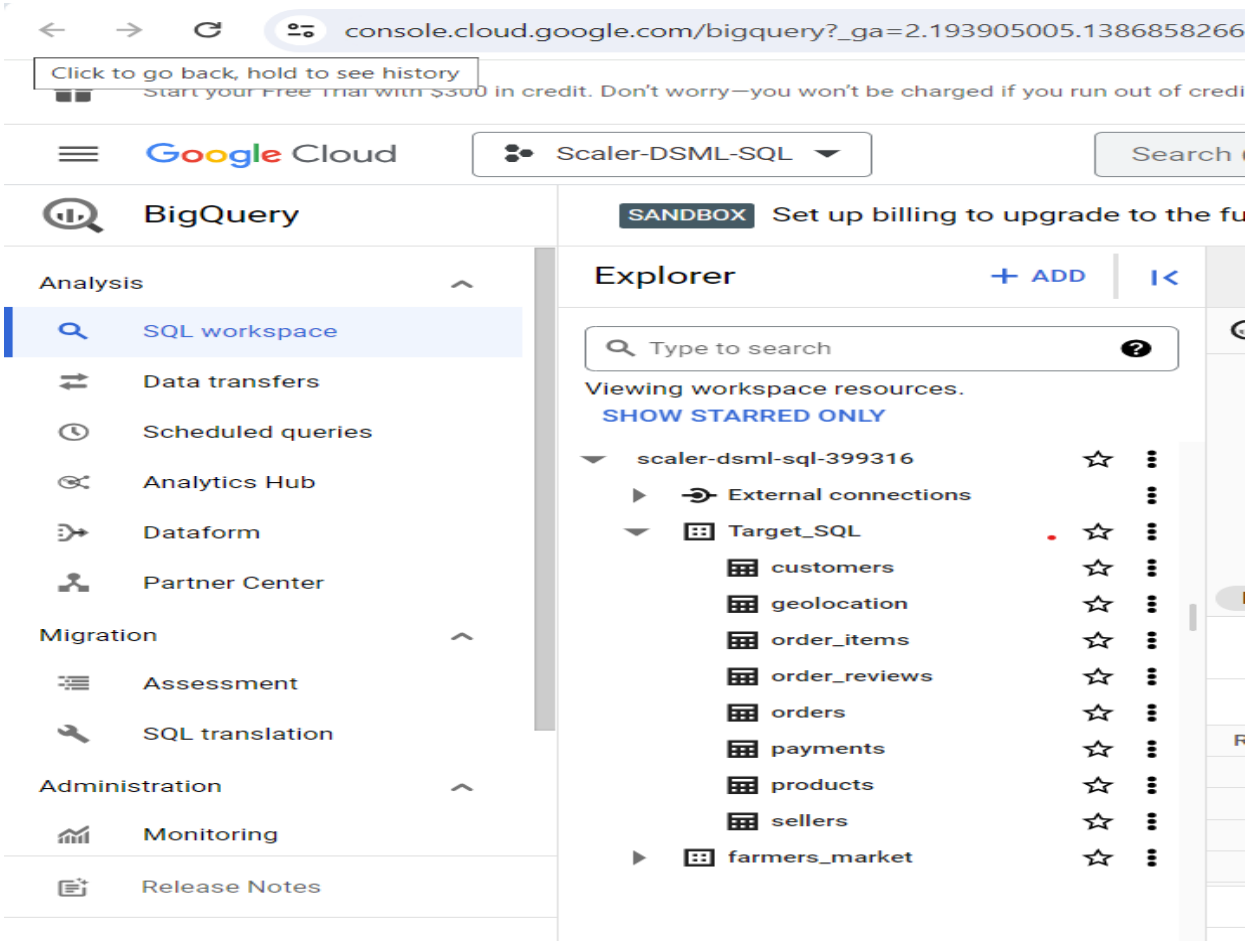
I am submitting the Business case study. If you need more clarity or any tips you want to provide for me, please connect with me on slack or WhatsApp (95003-78452) or on – call (99444-97897) or jeevajuly2020@gmail.com.

With the use of below tables, I have tried the Business case study:

csv files:

1. customers.csv
2. sellers.csv
3. order_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

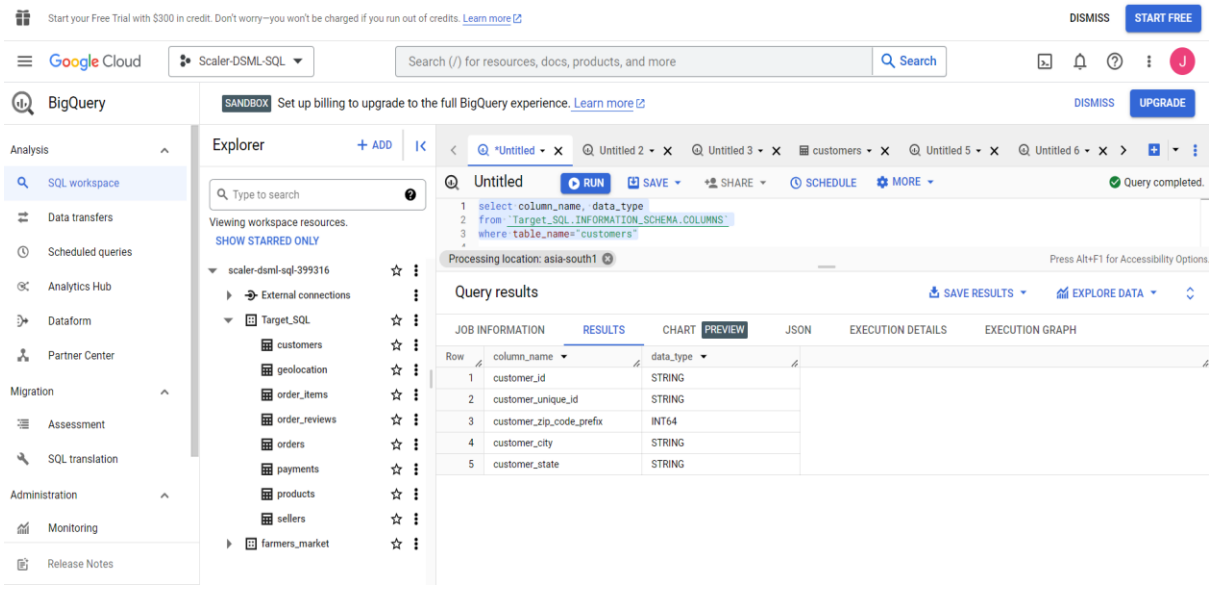
- ➔ I have uploaded all the files in Big Query platform.
- ➔ Created Data Set Name as Target_SQL.
- ➔ Table names same as CSV name. (customers, sellers, order_items, geolocation, payments, reviews, orders, products.)



Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

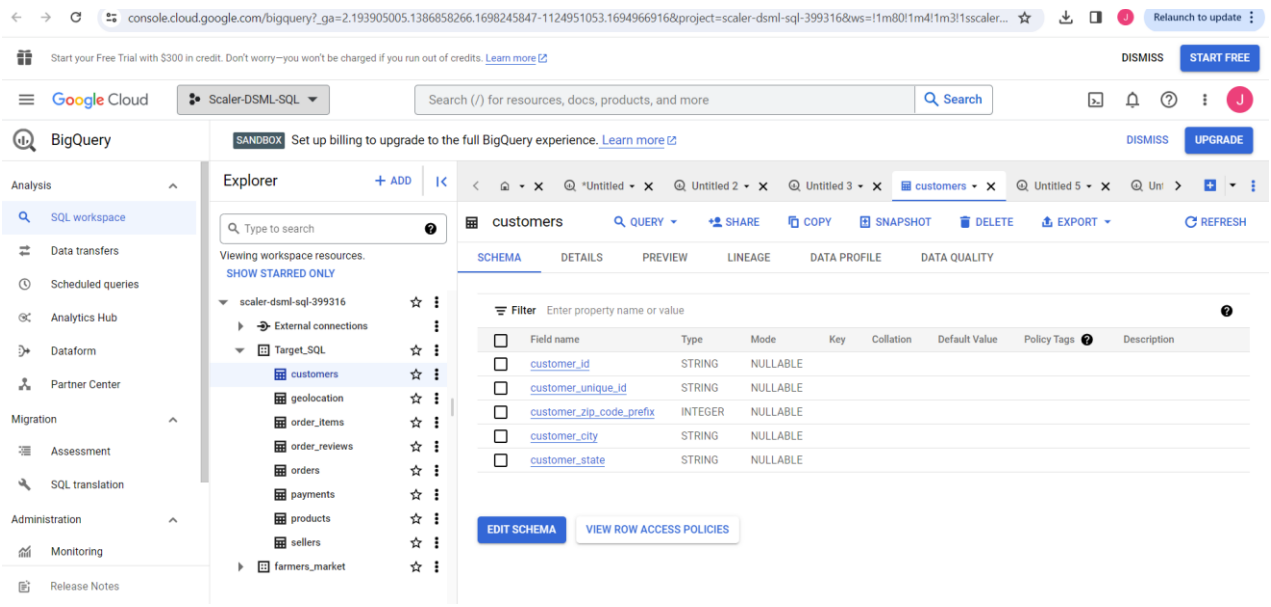
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
    - **A) Data type of all columns in the "customers" table.**

```
select column_name, data_type
from `Target_SQL.INFORMATION_SCHEMA.COLUMNS`
where table_name="customers"
```




We can extract the Data Type of “Customers” Table by above Query. We have to understand what the data type in the tables to analyse. May be in sometime, we need to convert the data type to the required data type.

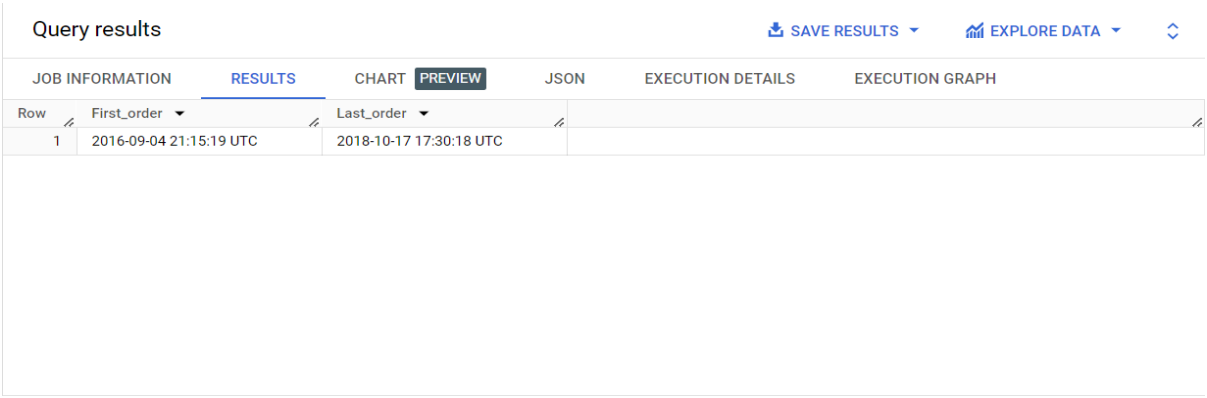
In Big Query, If we click on the table, it will automatically show the data type like below:



Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

### B) Get the time range between which the orders were placed.

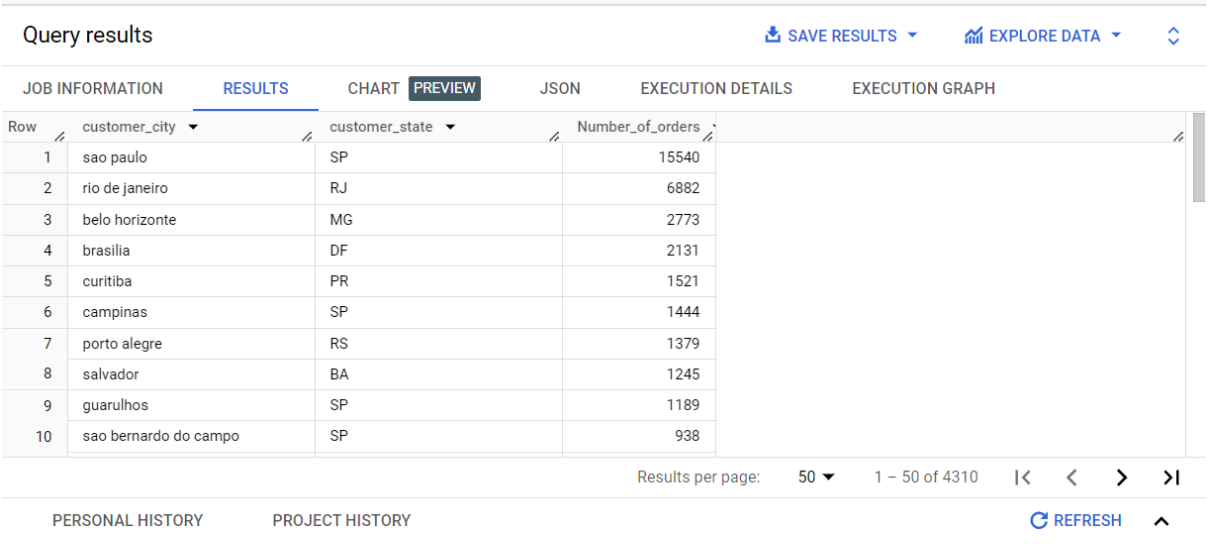
```
SELECT
  min(order_purchase_timestamp) First_order,
  max(order_purchase_timestamp) Last_order
FROM `Target_SQL.orders`
```




In the given data set, between 2016-09-04 21:15:19 UTC and 2018-10-17 17:30:18 UTC which orders were placed (i.e.… first order made in the E-commerce is on the timestamp 2016-09-04 21:15:19 UTC and the last order made in the E-commerce is on the timestamp 2018-10-17 17:30:18 UTC). During that duration only all the orders were made.

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

```
SELECT
  DISTINCT c.customer_city,
  c.customer_state,
  COUNT(o.customer_id) AS Number_of_orders
FROM `Target_SQL.orders` o
JOIN `Target_SQL.customers` c
ON o.customer_id = c.customer_id
WHERE TIMESTAMP(order_purchase_timestamp) BETWEEN TIMESTAMP('2016-09-04 21:15:19 UTC') AND
TIMESTAMP('2018-10-17 17:30:18 UTC')
GROUP BY c.customer_city, c.customer_state ORDER BY Number_of_orders DESC
```



Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

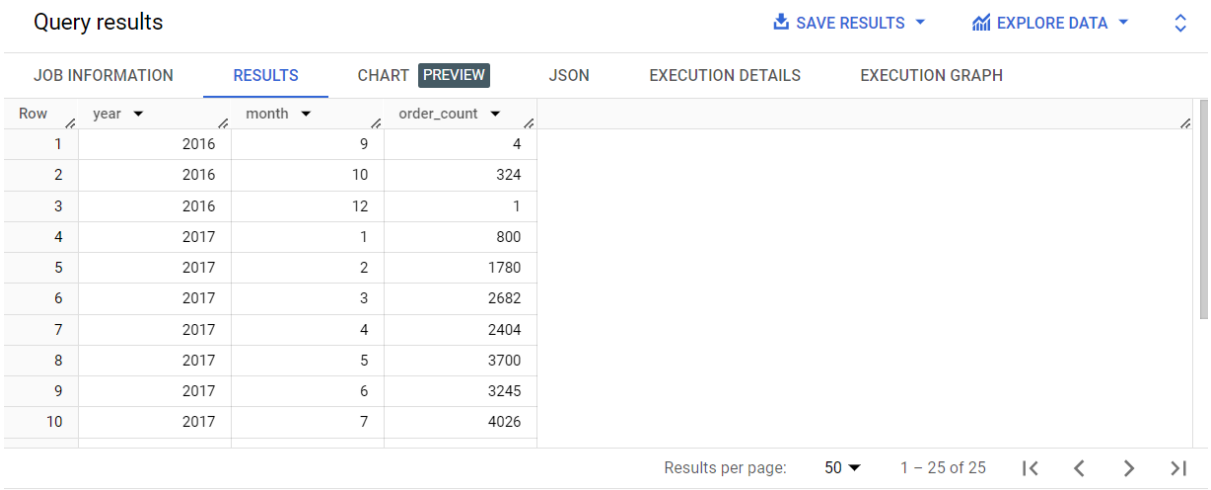
Here we are analysing the cities and states of customers who placed orders during specified time period. That's how customers are distributed for the business. Here, in the city Sao Paulo has more orders. We can consider that we have reached to the customer hugely there and our target customers are more there. For our business they are the richest consumers of the product.

Cities with less number of orders we have to do more marketing, campaigns, advertisements in pamphlet, online ads, etc.… to reach there more.

2. **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,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  COUNT(DISTINCT order_id) AS order_count
FROM `Target_SQL.orders`
GROUP BY year, month
ORDER BY year, month
```




| Row | year | 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 |

On seeing the data, we can apparently tell that there is growth in number of order. On comparing months in year we can get an idea in which month more number of orders are placed.

The results from this query can inform data-driven decision-making, helping businesses optimize their strategies based on historical order trends.

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

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  COUNT(DISTINCT order_id) AS order_count
FROM
  `Target_SQL.orders`
GROUP BY month
ORDER BY month
```

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | 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 |

Results per page: 50 1 – 12 of 12

This type of analysis can be valuable for inventory management, staffing, and marketing efforts. For example, businesses may need to adjust their operations during peak months to meet higher demand. We can see that March to August there good number of orders are placed. August reached the highest order made in that seasonality.

The results from this query can inform data-driven decision-making, helping businesses optimize their strategies based on historical order trends.

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

a. **0-6 hrs : Dawn**
b. **7-12 hrs : Mornings**
c. **13-18 hrs : Afternoon**
d. **19-23 hrs : Night**

```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 0 AND 5 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 6 AND 11 THEN 'Morning'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 12 AND 17 THEN 'Afternoon'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 18 AND 23 THEN 'Night'
  END AS time_of_the_day,
  COUNT(o.order_id) AS Number_of_order
FROM Target_SQL.orders o
JOIN Target_SQL.customers c
    ON o.customer_id = c.customer_id
GROUP BY time_of_the_day
ORDER BY Number_of_order DESC
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | time_of_the_day | Number_of_order |
|---|---|---|
| 1 | Afternoon | 38361 |
| 2 | Night | 34100 |
| 3 | Morning | 22240 |
| 4 | Dawn | 4740 |

PERSONAL HISTORY | PROJECT HISTORY | REFRESH

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

This query provides insights into when customers tend to place orders during the day. It can help businesses identify peak order times, optimize marketing strategies, and allocate resources effectively. Accordingly, Afternoon followed by Night Number of orders is high.

Understanding the time of day when orders are most frequent can inform decisions related to customer support staffing, order fulfilment scheduling, and targeted promotions.

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

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

```
SELECT
  c.customer_state,
  EXTRACT(month FROM o.order_purchase_timestamp) AS month,
  COUNT(o.order_purchase_timestamp) AS order_count
FROM Target_SQL.orders o
JOIN Target_SQL.customers c
 ON o.customer_id = c.customer_id
GROUP BY c.customer_state, month
ORDER BY c.customer_state, month
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | customer_state | 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 |
| 10 | AC | 10 | 6 |

Results per page: 50 1 – 50 of 322

This query helps identify any seasonal patterns in order placement, which can be valuable for planning marketing campaigns, inventory management, and staffing. By associating orders with specific states, businesses can gain insights into regional variations in ordering behaviour.

The results can guide businesses in making data-driven decisions related to staffing, and marketing strategies, adapting to changing order patterns.

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

```
SELECT
  customer_state,
  COUNT(distinct customer_id) AS no_of_customers
FROM `Target_SQL.customers`
GROUP BY customer_state
ORDER BY no_of_customers DESC
```

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | customer_state | no_of_customers |
|---|---|---|
| 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 |

Results per page: 50 | 1 – 27 of 27

This analysis provides valuable insights into the distribution of customers across different states. It helps identify states with a large customer base, which can be useful for business expansion, marketing strategies, and logistics planning.

The information obtained from this query can aid businesses in making data-driven decisions, such as where to focus marketing efforts, allocate resources, or open new retail locations.

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).
   You can use the "payment_value" column in the payments table to get the cost of orders.

```
WITH YearlyOrderCosts AS (
                        SELECT
                           EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
                           EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
                           SUM(p.payment_value) AS total_cost
                          FROM `Target_SQL.orders` o
                          JOIN `Target_SQL.payments` p
                            ON o.order_id = p.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 order_year, order_month
                            )
SELECT
  o.order_month,
  (o.total_cost - p.total_cost) / p.total_cost * 100 AS cost_increase_percentage
FROM YearlyOrderCosts o
JOIN YearlyOrderCosts p
  ON o.order_month = p.order_month
WHERE o.order_year = 2018 AND p.order_year = 2017
ORDER BY o.order_year, o.order_month
```

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | order_month | cost_increase_perce |
|---|---|---|
| 1 | 1 | 705.1266954171... |
| 2 | 2 | 239.9918145445... |
| 3 | 3 | 157.7786066709... |
| 4 | 4 | 177.8407701149... |
| 5 | 5 | 94.62734375677... |
| 6 | 6 | 100.2596912456... |
| 7 | 7 | 80.04245463390... |
| 8 | 8 | 51.60600520477... |

PERSONAL HISTORY | PROJECT HISTORY REFRESH

We have calculated the increase percentage in the cost of order from 2017 and 2018, only month from Jan to Aug. January has cost increase percentage followed by February and April. With this data we can understand increase in cost of order by 705.1266954171… which is January.

This analysis can assist business in understanding the cost trends over time and making informed decisions based on changes in order costs.

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

```
SELECT
  c.customer_state,
  Round (SUM(oi.price),2) AS total_order_price,
  Round (AVG(oi.price),2) AS average_order_price
FROM `Target_SQL.customers` c
JOIN `Target_SQL.orders` o
ON c.customer_id = o.customer_id
JOIN `Target_SQL.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | customer_state | total_order_price | average_order_price |
|---|---|---|---|
| 1 | AC | 15982.95 | 173.73 |
| 2 | AL | 80314.81 | 180.89 |
| 3 | AM | 22356.84 | 135.5 |
| 4 | AP | 13474.3 | 164.32 |
| 5 | BA | 511349.99 | 134.6 |
| 6 | CE | 227254.71 | 153.76 |
| 7 | DF | 302603.94 | 125.77 |
| 8 | ES | 275037.31 | 121.91 |
| 9 | GO | 294591.95 | 126.27 |
| 10 | MA | 119648.22 | 145.2 |

Results per page: 50 1 – 27 of 27

This analysis will provides the total order price and average order price for each state, which is the sum of prices for all orders made by customers in that particular state. State with higher total

order prices have more orders or higher value orders. States with higher average order prices may indicate customers in those states tend to purchase more expensive items or place larger orders.

This information can be valuable for business decision making like targeting specific states for marketing campaigns or adjusting pricing strategies based on reginal preferences.

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

```
SELECT
  c.customer_state,
  round(SUM(oi.freight_value), 2) AS total_order_freight,
  round(AVG(oi.freight_value), 2) AS average_order_freight
FROM `Target_SQL.customers` c
JOIN `Target_SQL.orders` o
ON c.customer_id = o.customer_id
JOIN `Target_SQL.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | customer_state | total_order_freight | average_order_freigh |
|---|---|---|---|
| 1 | AC | 3686.75 | 40.07 |
| 2 | AL | 15914.59 | 35.84 |
| 3 | AM | 5478.89 | 33.21 |
| 4 | AP | 2788.5 | 34.01 |
| 5 | BA | 100156.68 | 26.36 |
| 6 | CE | 48351.59 | 32.71 |
| 7 | DF | 50625.5 | 21.04 |
| 8 | ES | 49764.6 | 22.06 |
| 9 | GO | 53114.98 | 22.77 |
| 10 | MA | 31523.77 | 38.26 |

Results per page: 50 | 1 – 27 of 27

PERSONAL HISTORY | PROJECT HISTORY | REFRESH

By analysing total and average order freight values by state, businesses can identify regions with high shipping costs and explore opportunities for cost optimization. This data can also help in setting pricing or shipping policies for different states.

To gain deeper insights, businesses may consider visualizing this data or conducting additional analysis to understand the underlying factors contributing to variations in freight costs among states.

5. **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:

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

a. **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
b. **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

```
SELECT
order_id,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, DAY) AS time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS
estimated_delivery_in_days,
DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM `Target_SQL.orders`
WHERE DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS NOT NULL
AND DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) IS NOT
NULL
GROUP BY time_to_deliver, order_id, estimated_delivery_in_days, diff_estimated_delivery
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | order_id | time_to_deliver | estimated_delivery_in_days | diff_estimated_delivery |
|---|---|---|---|---|
| 1 | 1950d777989f6a877539f5379... | 30 | 17 | -12 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 30 | 59 | 28 |
| 3 | 65d1e226dfaeb8cdc42f66542... | 35 | 52 | 16 |
| 4 | 635c894d068ac37e6e03dc54e... | 30 | 32 | 1 |
| 5 | 3b97562c3aee8bdedcb5c2e45... | 32 | 33 | 0 |
| 6 | 68f47f50f04c4cb6774570cfde... | 29 | 31 | 1 |
| 7 | 276e9ec344d3bf029ff83a161c... | 43 | 39 | -4 |
| 8 | 54e1a3c2b97fb0809da548a59... | 40 | 36 | -4 |
| 9 | fd04fa4105ee8045f6a0139ca5... | 37 | 35 | -1 |
| 10 | 302bb8109d097a9fc6e9cefc5... | 33 | 28 | -5 |

Results per page: 50 | 1 – 50 of 96476

When deviations between estimated and actual delivery dates are significant, businesses can investigate the reasons for delays or early deliveries. This can lead to process improvements.

The analysis of delivery times and deviations can inform businesses about areas where they can optimize their logistics and supply chain operations. Meeting or exceeding estimated delivery times can lead to higher customer satisfaction, potentially resulting in repeat business.

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

```
WITH StateFreightAvg AS (
  SELECT
    c.customer_state,
    AVG(oi.freight_value) AS avg_freight_value
  FROM `Target_SQL.customers` c
  JOIN `Target_SQL.orders` o
  ON c.customer_id = o.customer_id
  JOIN `Target_SQL.order_items` oi
  ON o.order_id = oi.order_id
  GROUP BY c.customer_state
)
```

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

```
(SELECT
  customer_state,
  avg_freight_value
FROM StateFreightAvg
ORDER BY avg_freight_value
LIMIT 5)

UNION ALL

(SELECT
  customer_state,
  avg_freight_value
FROM StateFreightAvg
ORDER BY avg_freight_value DESC
LIMIT 5)
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | customer_state | avg_freight_value |
|---|---|---|
| 1 | SP | 15.14727539041... |
| 2 | PR | 20.53165156794... |
| 3 | MG | 20.63016680630... |
| 4 | RJ | 20.96092393168... |
| 5 | DF | 21.04135494596... |
| 6 | RR | 42.98442307692... |
| 7 | PB | 42.72380398671... |
| 8 | RO | 41.06971223021... |
| 9 | AC | 40.07336956521... |
| 10 | PI | 39.14797047970... |

The variation in average freight values between states might be attributed to factors such as distance, infrastructure, and logistics. These observations provide insights into the differences in average freight values among various states, which could be useful for optimizing shipping and logistics strategies for e-commerce businesses operating in Brazil.

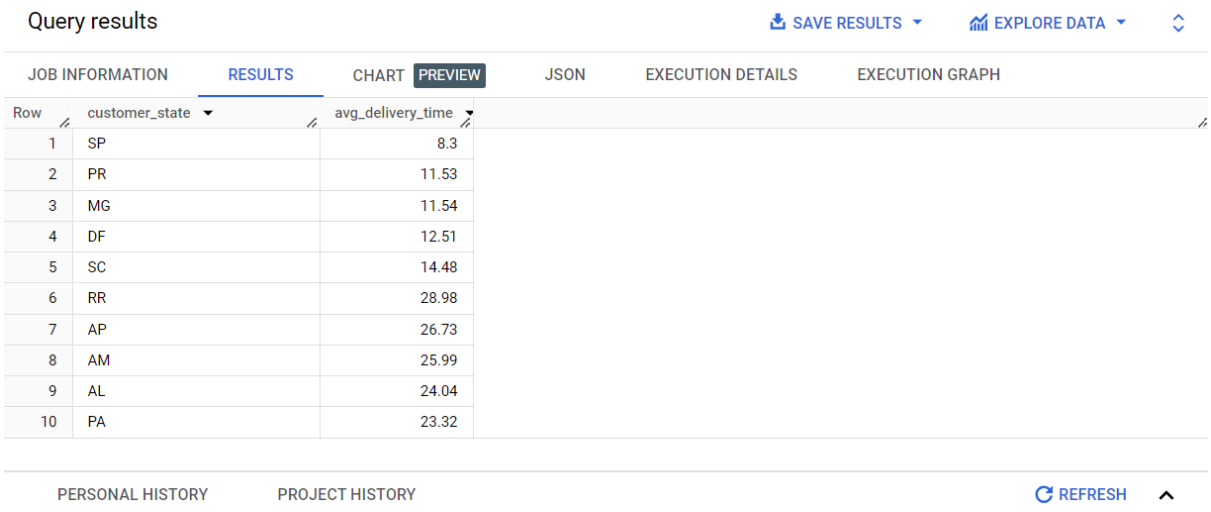
The data can help manage customer expectations. Customers in states with higher average freight costs may be more understanding of shipping charges, while those in areas with lower costs may have different expectations regarding shipping fees.

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

```
WITH avgdeliverytime as

(SELECT
customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)), 2)
AS avg_delivery_time
FROM`Target_SQL.orders` o
JOIN`Target_SQL.customers` c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
order by avg_delivery_time)

(SELECT
  customer_state,
  avg_delivery_time
```

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

```
FROM avgdeliverytime
ORDER BY avg_delivery_time
LIMIT 5)

UNION ALL

(SELECT
  customer_state,
  avg_delivery_time
FROM avgdeliverytime
ORDER BY avg_delivery_time desc
LIMIT 5)
```




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

SP has the lowest average delivery time, indicating efficient logistics and infrastructure. Customers in SP experience faster order deliveries. PR, MG, and DF: These states also have relatively low average delivery times, suggesting well-organized supply chains and transportation networks. SC SC rounds up the list of states with the fastest average delivery times, indicating a strong e-commerce infrastructure. RR has the highest average delivery time, which may attribute to its remote location and transportation. AP, AM, AL & PA states have higher average delivery times, influenced by logistical factors.

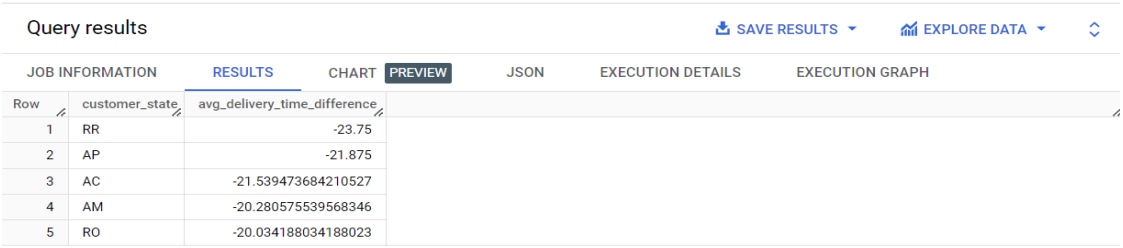
The data highlights the regional disparities in delivery times across different states in Brazil. Efficient logistics and transportation networks play a crucial role in faster deliveries. States with faster delivery times are likely to offer better customer experiences and higher satisfaction levels. E-commerce companies can leverage these insights to improve services in other regions. For states with longer delivery times, there are opportunities for logistics optimization to enhance customer satisfaction and compete effectively in those regions.

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.

```
SELECT
customer_state,
AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date, DAY)) AS
avg_delivery_time_difference
```

Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

```
FROM `Target_SQL.orders` o
JOIN `Target_SQL.customers` c ON o.customer_id = c.customer_id
WHERE DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date, DAY) < 0
AND order_status = 'delivered'
GROUP BY customer_state
ORDER BY avg_delivery_time_difference
limit 5
```



Query results — SAVE RESULTS — EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

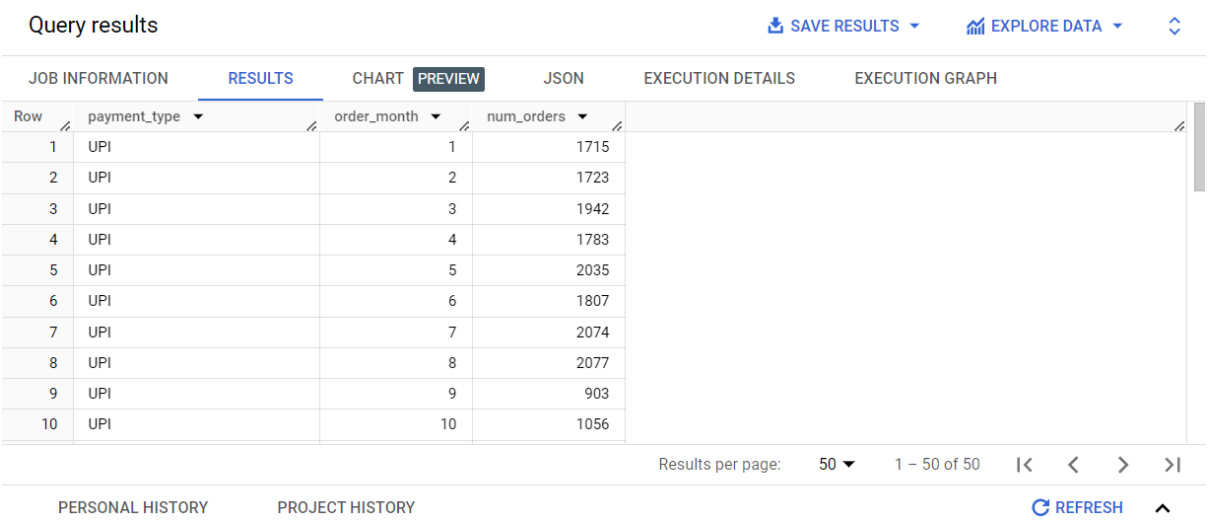
| Row | customer_state | avg_delivery_time_difference |
|---|---|---|
| 1 | RR | -23.75 |
| 2 | AP | -21.875 |
| 3 | AC | -21.539473684210527 |
| 4 | AM | -20.280575539568346 |
| 5 | RO | -20.034188034188023 |

Above states have a significant negative difference, indicating that, on average, orders are delivered well ahead of their estimated delivery dates, making them efficient in terms of order delivery.

The impact of efficient order delivery extends to customer satisfaction, business success, competitive advantage, and economic growth. It benefits not only e-commerce platforms but also the regions where these efficient services are provided.

6. **Analysis based on the payments.**
   A) Find the month on month no. of orders placed using different payment types.

```
SELECT
p.payment_type,
EXTRACT(MONTH FROM TIMESTAMP(o.order_purchase_timestamp)) AS order_month,
COUNT(DISTINCT o.order_id) AS num_orders
FROM `Target_SQL.orders` o
JOIN `Target_SQL.payments` p
ON o.order_id = p.order_id
GROUP BY  payment_type, order_month
ORDER BY  payment_type, order_month
```



Query results — SAVE RESULTS — EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | payment_type | order_month | num_orders |
|---|---|---|---|
| 1 | UPI | 1 | 1715 |
| 2 | UPI | 2 | 1723 |
| 3 | UPI | 3 | 1942 |
| 4 | UPI | 4 | 1783 |
| 5 | UPI | 5 | 2035 |
| 6 | UPI | 6 | 1807 |
| 7 | UPI | 7 | 2074 |
| 8 | UPI | 8 | 2077 |
| 9 | UPI | 9 | 903 |
| 10 | UPI | 10 | 1056 |

Results per page: 50 — 1 – 50 of 50

PERSONAL HISTORY | PROJECT HISTORY — REFRESH

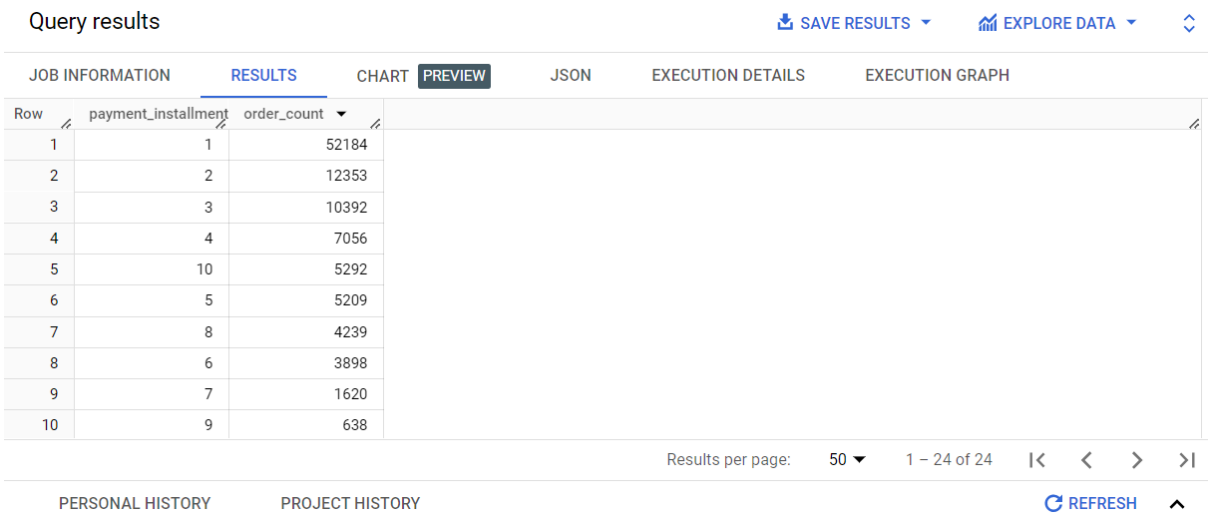
Jeeva Muthukumar K
99444-97897/ jeevajuly2020@gmail.com

The query provides insights into payment method trends over time, showing how customers prefer to pay for their orders each month. It allows businesses to identify any seasonal patterns in payment method usage and understand which methods are more popular during specific months.

Businesses can align marketing campaigns and promotions with the most favoured payment methods during different months, enhancing customer engagement. Identifying changing payment trends helps in providing a seamless payment experience for customers, leading to higher satisfaction. Companies can allocate resources effectively to ensure smooth payment processing for the most used methods, enhancing efficiency. Understanding shifting payment preferences allows companies to assess and manage potential risks associated with specific methods. Businesses can manage finances more effectively and optimize resources based on payment trends.

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

```
SELECT
p.payment_installments,
COUNT(o.order_id) AS order_count
FROM `Target_SQL.orders` o
JOIN `Target_SQL.payments` p
ON o.order_id = p.order_id
WHERE o.order_status != 'canceled'
GROUP BY p.payment_installments
ORDER BY order_count desc
```




The table shows the number of orders associated with different payment installment options. The data is sorted in descending order of order count.

The data provides valuable insights that can help businesses optimize their payment options and improve financial planning and fraud detection.