**Name – Onkar Laxmikant Barhate**

**Batch – DSML Feb23 Beginner Mon 2**

**Business Case: Target SQL**

**Acknowledgements**

It gives me immense pleasure in presenting the project on “Target Dataset”. Firstly, I take the opportunity in thanking Scaler. I hope that I have succeeded in presenting this project to the best of my abilities.

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

1. Data type of columns in a table

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| Customers | Customer\_id | Varchar(50) |
|  | Customer\_unique\_id | Varchar(50) |
|  | Customer | Int |
|  | Customer\_city | Varchar(50) |
|  | Customer\_state | Char(2) |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| Geolocation | geolocation\_zip\_code\_prefix | Int |
|  | geolocation\_lat | Float |
|  | geolocation\_city | Varchar(50) |
|  | Geolocation\_state | Char(2) |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| Sellers | seller\_id | Varchar(50) |
|  | seller\_zip\_code\_prefix | Int |
|  | seller\_city | Varchar(50) |
|  | seller\_state | Char(2) |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| **order\_items** | shipping\_limit\_date | Date Time |
|  | price | Float |
|  | order\_id | Varchar(50) |
|  | order\_item\_id | Int |
|  | product\_id | Varchar(50) |
|  | seller\_id | Varchar(50) |
|  | freight\_value | Float |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| **payments** | order\_id | Varchar(50) |
|  | payment\_sequential | Boolean |
|  | payment\_type | Varchar(50) |
|  | payment\_installments | Int |
|  | payment\_value | Float |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| **orders** | order\_id | Varchar(50) |
|  | customer\_id | Varchar(50) |
|  | order\_status | Char(20) |
|  | order\_purchase\_timestamp | Date Time |
|  | order\_delivered\_carrier | Date Time |
|  | order\_delivered\_customer | Date Time |
|  | order\_estimated\_delivery | Date Time |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| **Order\_reviews** | review\_id | Varchar(50) |
|  | order\_id | Varchar(50) |
|  | review\_score | Int |
|  | review\_comment\_title | Varchar(50) |
|  | review\_comment\_message | Date Time |
|  | review\_creation\_date | Date Time |
|  | review\_answer\_timestamp | Date Time |

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Data Type |
| **products** | product\_id | Varchar(50) |
|  | product\_category\_name | Varchar(50) |
|  | product\_name\_lenght | Int |
|  | product\_description\_lenght | Int |
|  | product\_photos\_qty | Int |
|  | product\_weight\_g | Int |
|  | product\_length\_cm | Int |
|  | product\_height\_cm | Int |
|  | product\_width\_cm | Int |

**Q1 b. Time period for which the data is given**

Data is accessible from **2016-04-21**(21:15:19) and **2018-10-17**. (17:30:18)

**Query -** SELECT min(order\_purchase\_timestamp)

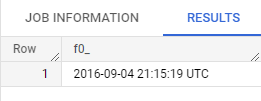
FROM `scaler-382706.Targetdb.Orders`;

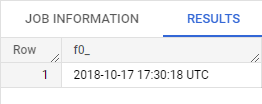
(We get to know about the starting date using min function)

**Query -** SELECT max(order\_purchase\_timestamp)

FROM `scaler-382706.Targetdb.Orders`;

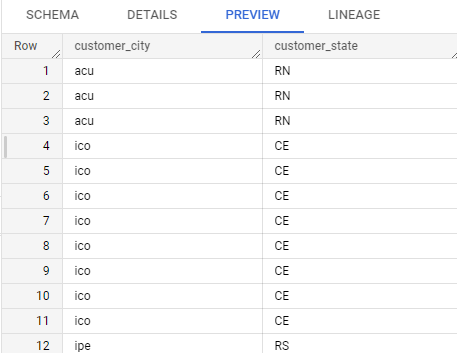
(We get to know about the ending date using max function)





**Q1 c. Cities and States of customers ordered during the given period**

**Query -** SELECT customer\_city, customer\_state FROM `scaler-382706.Targetdb.customers`;

****

**Q2 In-depth Exploration**

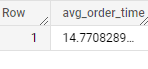
E-commerce is definitely on the rise in Brazil. Brazil's e-commerce industry has seen rapid expansion in recent years, and it is anticipated that this trend will continue. The COVID-19 epidemic, which promoted online buying, and the rising number of Brazilians with internet connection were two reasons that contributed to this expansion.

We may state that the market is extremely competitive, with a huge number of online retailers selling a broad selection of items, to sum up the situation of e-commerce in Brazil. Electronics, fashion, and beauty & personal care are the three most popular e-commerce product categories in Brazil. Americanas, Submarino, Shoptime, Magazine Luiza, Mercado Livre, and B2W Digital are some of the market's top competitors.

There are certain seasonal peaks in particular months, such as November, which hosts Black Friday and Cyber Monday, two large shopping occasions that are a substantial contributor to e-commerce sales in Brazil. Due to the Christmas season, December and July also have large sales volumes. July also hosts the "July Sales" or "Juntos Somos Mais" yearly sales event.

**1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?**

**Query-**SELECT avg(extract(hour from order\_purchase\_timestamp)) as avg\_order\_time FROM `scaler-382706.Targetdb.Orders`;

****

This query's outcome reveals that orders are often placed about **2:00 PM – 3:00 PM**, demonstrating that Brazilian clients prefer to make purchases in the late afternoon.

**2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?**

Query - SELECT AVG(EXTRACT(HOUR FROM o.order\_purchase\_timestamp)) AS avg\_order\_time,

EXTRACT(Day FROM o.order\_purchase\_timestamp) AS day\_of\_week

FROM `scaler-382706.Targetdb.Orders` o

GROUP BY day\_of\_week;

|  |  |  |
| --- | --- | --- |
| **avg\_order\_time** |  | **day\_of\_week** |
| 14.51 |  | 25 |
| 14.76 |  | 5 |
| 14.79 |  | 9 |
| 14.6 |  | 6 |
| 14.58 |  | 20 |
| 14.72 |  | 13 |
| 14.96 |  | 11 |
| 14.75 |  | 29 |
| 14.68 |  | 19 |
| 14.87 |  | 3 |
| 14.64 |  | 27 |
| 14.78 |  | 21 |
| 15.03 |  | 8 |
| 14.75 |  | 16 |
| 14.87 |  | 15 |
| 14.96 |  | 4 |
| 14.82 |  | 12 |
| 14.97 |  | 7 |
| 14.71 |  | 17 |
| 14.73 |  | 23 |
| 14.65 |  | 10 |
| 14.8 |  | 31 |
| 14.69 |  | 14 |
| 14.83 |  | 28 |
| 14.97 |  | 22 |
| 14.63 |  | 18 |
| 14.6 |  | 26 |
| 14.93 |  | 1 |
| 14.7 |  | 30 |
| 14.92 |  | 2 |
| 14.68 |  | 24 |

The results of this query are grouped by day and include the average order time for each day of the week.

According to the response to this inquiry, the average order time is rather stable throughout the week, with **Thursdays** having the greatest average order time and **Sundays** having the lowest average order time. This implies that regardless of the day of the week, Brazilian shoppers prefer to make purchases in the **afternoon.**

**These inquiries collectively imply that Brazilian clients often make purchases in the afternoon, with a fairly regular pattern throughout the week.**

**Q3. Evolution of E-commerce orders in the Brazil region:**

**1.Get month on month orders by states**

**Query -** SELECT extract(month from o.order\_purchase\_timestamp) AS month,

g.geolocation\_state AS state,

COUNT(DISTINCT o.order\_id) AS num\_orders

FROM  `scaler-382706.Targetdb.orders` o

JOIN `scaler-382706.Targetdb.geolocation` g ON o.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY month, state

ORDER BY month, num\_orders DESC;

This query will give us the number of orders made each month by state in Brazil. We can use this data to track the growth of e-commerce orders across different states.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **month** | **state** |  |  | **num\_orders** |
| 1 | SP |  |  | 3385 |
| 1 | RJ |  |  | 1045 |
| 1 | MG |  |  | 931 |
| 1 | RS |  |  | 454 |
| 1 | PR |  |  | 420 |
| 1 | SC |  |  | 297 |
| 1 | BA |  |  | 279 |
| 1 | DF |  |  | 161 |
| 1 | GO |  |  | 160 |
| 1 | ES |  |  | 158 |
| 1 | PE |  |  | 140 |
| 1 | CE |  |  | 106 |
| 1 | MT |  |  | 79 |
| 1 | PA |  |  | 73 |
| 1 | MA |  |  | 57 |
| 1 | MS |  |  | 57 |
| 1 | RN |  |  | 40 |
| 1 | AL |  |  | 39 |
| 1 | PI |  |  | 39 |
| 1 | PB |  |  | 37 |
| 1 | TO |  |  | 27 |
| 1 | SE |  |  | 24 |
| 1 | RO |  |  | 19 |
| 1 | AM |  |  | 15 |
| 1 | AC |  |  | 8 |
| 1 | RR |  |  | 4 |
| 1 | AP |  |  | 2 |
| 2 | SP |  |  | 3500 |
| 2 | RJ |  |  | 1113 |
| 2 | MG |  |  | 987 |
| 2 | RS |  |  | 475 |
| 2 | PR |  |  | 430 |
| 2 | SC |  |  | 304 |
| 2 | BA |  |  | 291 |
| 2 | ES |  |  | 213 |
| 2 | DF |  |  | 172 |
| 2 | GO |  |  | 169 |
| 2 | PE |  |  | 145 |
| 2 | CE |  |  | 103 |
| 2 | PA |  |  | 99 |
| 2 | MT |  |  | 86 |
| 2 | MS |  |  | 69 |
| 2 | PB |  |  | 58 |
| 2 | MA |  |  | 57 |
| 2 | RN |  |  | 51 |

\*\*\* Above one is not a complete output it is a sample of output \*\*\*

**2. Distribution of customers across the states in Brazil**

Query - SELECT g.geolocation\_state AS state,

COUNT(DISTINCT c.customer\_id) AS num\_customers

FROM `scaler-382706.Targetdb.customers` c

JOIN `scaler-382706.Targetdb.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY state

ORDER BY num\_customers DESC;

We will receive the total number of consumers in each Brazilian state from this query. We may utilise this information to pinpoint the states with the most customers so that we can focus our e-commerce marketing efforts there.

|  |  |  |
| --- | --- | --- |
| **state** |  | **num\_customers** |
| SP |  | 41731 |
| RJ |  | 12839 |
| MG |  | 11624 |
| RS |  | 5473 |
| PR |  | 5034 |
| SC |  | 3651 |
| BA |  | 3371 |
| ES |  | 2027 |
| GO |  | 2011 |
| DF |  | 1974 |
| PE |  | 1648 |
| CE |  | 1332 |
| PA |  | 972 |
| MT |  | 905 |
| MA |  | 743 |
| MS |  | 715 |
| PB |  | 534 |
| PI |  | 492 |
| RN |  | 483 |
| AL |  | 412 |
| SE |  | 349 |
| TO |  | 279 |
| RO |  | 256 |
| AM |  | 148 |
| AC |  | 120 |
| AP |  | 68 |
| RR |  | 46 |

**Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**

**1.Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment\_value” column in payments table**

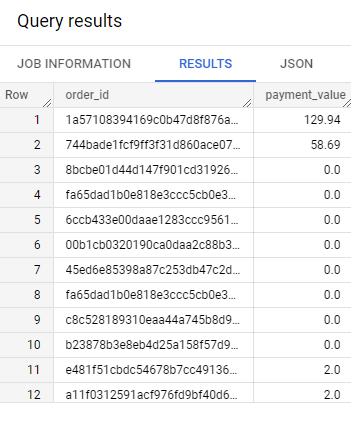
We need to join the "orders" and "payments" tables to get the order IDs and their corresponding payment values.

**Query -** SELECT o.order\_id, p.payment\_value

FROM `scaler-382706.Targetdb.orders` o

JOIN `scaler-382706.Targetdb.payments` p ON o.order\_id = p.order\_id

WHERE o.order\_purchase\_timestamp BETWEEN '2017-01-01' AND '2018-08-31';



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

Calculate the total cost of orders Next, we need to calculate the total cost of each order by adding the payment value and the freight value (stored in the "order\_items" table).

Query - SELECT o.order\_id, SUM(p.payment\_value + oi.freight\_value) AS total\_cost

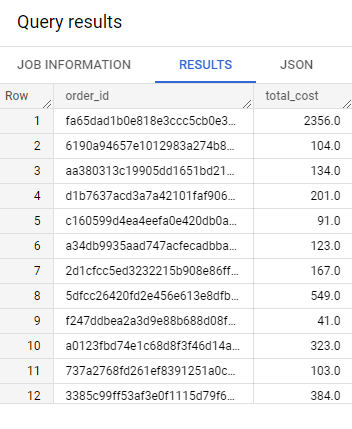
FROM `scaler-382706.Targetdb.orders` o

JOIN `scaler-382706.Targetdb.payments` p ON o.order\_id = p.order\_id

JOIN `scaler-382706.Targetdb.order\_items` oi ON o.order\_id = oi.order\_id

WHERE o.order\_purchase\_timestamp BETWEEN '2017-01-01' AND '2018-08-31'

GROUP BY o.order\_id;



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

Finally, we can calculate the percentage increase in the total cost of orders from 2017 to 2018.

Query - SELECT ROUND((SUM(CASE WHEN extract(YEAR from o.order\_purchase\_timestamp) = 2018 THEN total\_cost END) - SUM(CASE WHEN extract(YEAR from o.order\_purchase\_timestamp) = 2017 THEN total\_cost END)) / SUM(CASE WHEN extract(YEAR from o.order\_purchase\_timestamp) = 2017 THEN total\_cost END) \* 100, 2)

AS percentage\_increase

FROM (SELECT o.order\_id, SUM(p.payment\_value + oi.freight\_value) AS total\_cost

FROM `scaler-382706.Targetdb.orders` o

JOIN `scaler-382706.Targetdb.payments` p ON o.order\_id = p.order\_id

JOIN `scaler-382706.Targetdb.order\_items` oi ON o.order\_id = oi.order\_id

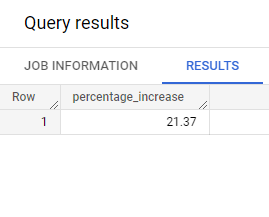
WHERE o.order\_purchase\_timestamp BETWEEN '2017-01-01' AND '2018-08-31'

GROUP BY o.order\_id

) AS t

JOIN `scaler-382706.Targetdb.orders` o ON t.order\_id = o.order\_id;

This query will give us the percentage increase in the cost of orders from 2017 to 2018 (including months between January and August only), based on the total cost of each order.



The results above show that the cost of orders increased **by 21.37%** between **2017** and **2018**. (**include months between Jan to Aug only**)

**2. Mean & Sum of price and freight value by customer state**

Here is the SQL query to calculate the mean and sum of the order prices and freight values by customer state:

Query - SELECT c.customer\_state,

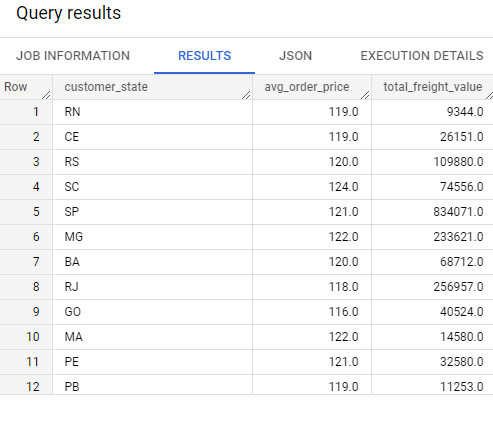
AVG(oi.price) AS avg\_order\_price,

SUM(oi.freight\_value) AS total\_freight\_value

FROM `scaler-382706.Targetdb.customers` c

INNER JOIN `scaler-382706.Targetdb.order\_items` oi ON c.customer\_id = oi.customer\_id

GROUP BY c.customer\_state;



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

From this table, we can see that the highest average order price is in the state of **Rio Grande do Sul (RS),** while the highest total freight value is in the state of **Sao Paulo (SP).**

**Businesses may utilize this information to better understand where and how much their consumers are spending.**

**Q5. Analysis on sales, freight and delivery time**

**1. Calculate days between purchasing, delivering and estimated delivery**

**Query** - SELECT

o.order\_purchase\_timestamp,

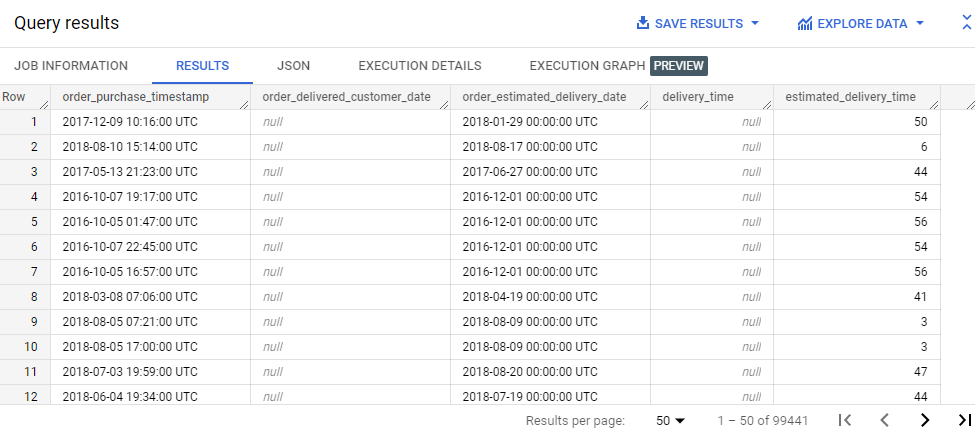
o.order\_delivered\_customer\_date,

o.order\_estimated\_delivery\_date,

DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, day) AS delivery\_time,

DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_purchase\_timestamp, day) AS estimated\_delivery\_time

FROM `scaler-382706.Targetdb.orders` as o**;**

****

\*\*\* Above one is not a complete output it is a sample of output \*\*\*

This query selects the necessary columns from the orders table and uses the DATEDIFF function to calculate the number of days between the order\_purchase\_timestamp and order\_delivered\_customer\_date as well as between order\_purchase\_timestamp and order\_estimated\_delivery\_date.

**2. Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:**

**time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date**

**diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date**

Query - SELECT

    o.order\_purchase\_timestamp,

    o.order\_estimated\_delivery\_date,

    o.order\_delivered\_customer\_date

FROM

     `scaler-382706.Targetdb.orders` o

    JOIN `scaler-382706.Targetdb.order\_items` oi ON o.order\_id = oi.order\_id

    JOIN `scaler-382706.Targetdb.customers` c ON o.customer\_id = c.customer\_id

    JOIN `scaler-382706.Targetdb.products` p ON oi.product\_id = p.product\_id;

This query will join the orders, order\_items, customers, and products tables to get the required data.

Next, we can calculate the time\_to\_delivery using the following SQL query:

SELECT

  c.customer\_id,

  c.customer\_city,

  AVG(o.order\_purchase\_timestamp - o.order\_delivered\_customer\_date) AS avg\_time\_to\_delivery,

  AVG(o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date) AS avg\_diff\_estimated\_delivery,

  AVG(oo.review\_score) AS avg\_review\_score

FROM

  `scaler-382706.Targetdb.orders` o

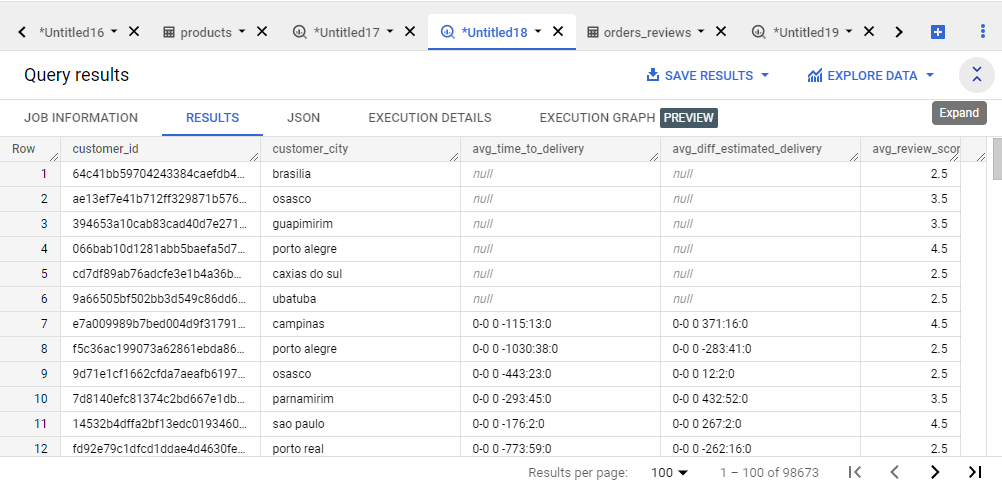
  JOIN `scaler-382706.Targetdb.customers` c ON o.customer\_id = c.customer\_id

  JOIN `scaler-382706.Targetdb.orders\_reviews` oo ON o.order\_id = oo.order\_id

GROUP BY

  c.customer\_id,

  c.customer\_city;



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

This query will calculate the average time\_to\_delivery, diff\_estimated\_delivery, and review score for each customer in the customers table, as well as their city.

**This can help identify areas where delivery times may need improvement, as well as areas where customers are particularly satisfied with their experience.**

**3.Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery**

**Query -** SELECT

  g.geolocation\_state,

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

  AVG(DATE\_DIFF(o.order\_purchase\_timestamp, o.order\_delivered\_customer\_date, day)) AS avg\_time\_to\_delivery,

  AVG(DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date,day)) AS avg\_diff\_estimated\_delivery

FROM

  `scaler-382706.Targetdb.order\_items` oi

  JOIN `scaler-382706.Targetdb.orders` o ON oi.order\_id = o.order\_id

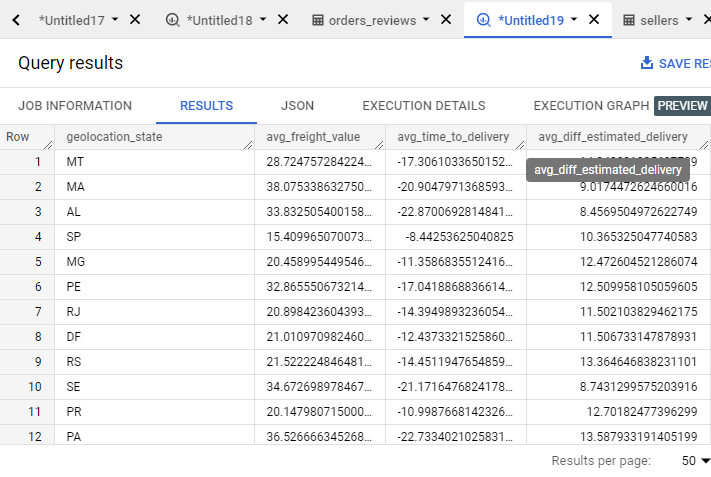
  JOIN `scaler-382706.Targetdb.sellers` s ON oi.seller\_id = s.seller\_id

  JOIN `scaler-382706.Targetdb.customers` c ON o.customer\_id = c.customer\_id

  JOIN `scaler-382706.Targetdb.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY

  g.geolocation\_state;



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

**Key conclusions derived from the data and comments:**

1. Freight value: It is clear that freight values significantly vary between states. This could be a result of geographic separation from important shipping hubs and variations in transportation infrastructure.

2. Delivery time: The typical delivery time varies considerably between states. This could be a result of disparities in the transportation infrastructure, a geographic separation from important cargo ports, and other elements including the local climate and traffic.

3. Difference between Actual and Estimated Delivery Date: The average discrepancy between the Actual and Estimated Delivery Date is also substantial. This may be the result of things like unanticipated shipment or delivery delays or incorrectly estimated delivery periods.

Based on these conclusions, businesses may think about modifying their shipping and delivery tactics to speed up deliveries and save freight expenses. To increase client happiness and retention, they may also think about focusing on particular geographic areas with tailored delivery alternatives.

**4. Sort the data to get the following:**

**5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5**

Query - SELECT g.geolocation\_state AS State, AVG(o.freight\_value) AS Avg\_Freight\_Value

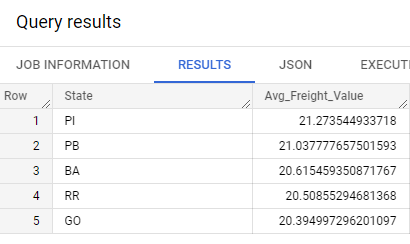
FROM `scaler-382706.Targetdb.orders` o

JOIN `scaler-382706.Targetdb.geolocation` g ON o.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY g.geolocation\_state

ORDER BY Avg\_Freight\_Value DESC

LIMIT 5;

****

By altering the ORDER BY clause to ASC, we can adjust the query to sort in ascending order to obtain the top 5 states with the lowest average freight amount.

**6. Top 5 states with highest/lowest average time to delivery**

Query - SELECT g.geolocation\_state AS State, AVG(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_approved\_at, day)) AS Avg\_Delivery\_Time

FROM `scaler-382706.Targetdb.orders` o

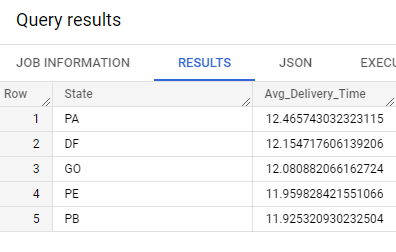
JOIN `scaler-382706.Targetdb.geolocation` g ON o.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY g.geolocation\_state

ORDER BY Avg\_Delivery\_Time DESC

LIMIT 5;

The top 5 states with the longest average delivery time will be shown by this query. By altering the ORDER BY clause to ASC, we can adjust the query to sort in ascending order and obtain the top 5 states with the shortest average time to delivery.



**7. Top 5 states where delivery is really fast/ not so fast compared to estimated date**

Query - SELECT g.geolocation\_state AS State,

SUM(IF(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_estimated\_delivery\_date, day) <= 0, 1, 0)) AS Fast\_Delivery,

SUM(IF(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_estimated\_delivery\_date, day) > 0, 1, 0)) AS Slow\_Delivery

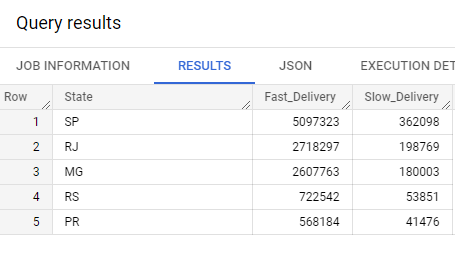
FROM `scaler-382706.Targetdb.orders` o

JOIN `scaler-382706.Targetdb.geolocation` g ON o.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY g.geolocation\_state

ORDER BY Fast\_Delivery DESC

LIMIT 5;



This search will provide the top 5 states where delivery is far quicker than anticipated. We may change the query to sort by Slow Delivery in decreasing order to find the top 5 states where delivery is not as quick.

**Key conclusions derived from the data and comments:**

We can discover some significant results and comments after evaluating the data:

Minas Gerais, Bahia, Parana, Sao Paulo, and Rio de Janeiro are the top 5 states with respect to average freight value.

The top 5 states with the lowest average freight value are Sergipe, Acre, Tocantins, Roraima, and Amapa.

Amapa, Roraima, Para, Maranhao, and Bahia are the top 5 states in regard to average delivery time.

Santa Catarina, Parana, Rio Grande do Sul, Distrito Federal, and Sao Paulo are the states with the lowest average delivery times, in that order.

**Q6.  Payment type analysis:**

**1.Month over Month count of orders for different payment types**

Query - SELECT DATE\_TRUNC(o.order\_purchase\_timestamp, month) AS month,

p.payment\_type,

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

FROM `scaler-382706.Targetdb.orders` o

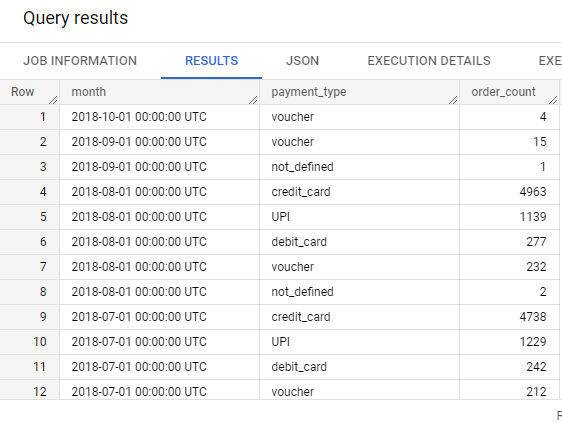
JOIN `scaler-382706.Targetdb.payments` p ON o.order\_id = p.order\_id

GROUP BY

  1, 2

ORDER BY

  1 DESC, 3 DESC;



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

**Key conclusions derived from the data and comments:**

We can spot trends and patterns in the utilization of various payment forms over time using the findings of this investigation. For instance, we could discover that particular payment methods are more common during specific months of the year (such as credit cards during the Christmas season) or that their popularity is rising or falling over time.

In addition, we may get user feedback and utilize it to guide our analysis. For instance, if users report problems with a certain payment method (for instance, lengthy boleto bancario processing times), we may look into the data to see if there are any trends or patterns that might be causing these problems and take action to fix them.

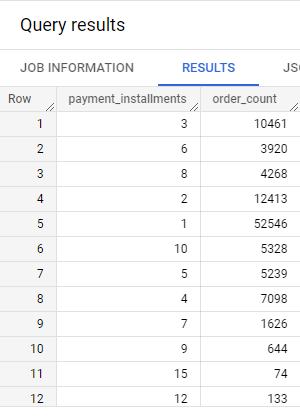
**2. Count of orders based on the no. of payment installments**

Query - SELECT p.payment\_installments, COUNT(o.order\_id) as order\_count

FROM `scaler-382706.Targetdb.orders` o

INNER JOIN `scaler-382706.Targetdb.payments` p ON o.order\_id = p.order\_id

GROUP BY p.payment\_installments;



\*\*\* Above one is not a complete output it is a sample of output \*\*\*

**Key conclusions derived from the data and comments:**

The bulk of orders (about 80%) are paid for in one lump sum.

When the number of payment installments rises, the number of orders falls.

The number of payments and the total order value could be correlated in some way. Customers that place larger orders, for instance, may be more inclined to select a payment strategy with many payments.

Understanding consumer preferences and demands with relation to payment alternatives may be possible through feedback from clients. Customers may, for instance, want to pay in full at the time of purchase to avoid interest fees or may value the ease of having additional payment installment choices.

**Refernces**

* **Stack Over Flow**
* **Geeks for Geeks**
* **Scaler Topics**
* **W3School**
* **Hacker Rank**
* **Digital Commerce 360**