**Business Case: Target SQL**

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***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | S.No. | Table | Columns | Datatype | | 1 | columns | customer\_id | Varchar(50) | | 2 | columns | customer\_unique\_id | Varchar(50) | | 3 | columns | customer\_zip\_code\_prefix | Int | | 4 | columns | customer\_city | Varchar(50) | | 5 | columns | customer\_state | Varchar(50) | | 6 | sellers | seller\_id | Varchar(50) | | 7 | sellers | seller\_zip\_code\_prefix | Varchar(50) | | 8 | sellers | seller\_city | Varchar(50) | | 9 | sellers | seller\_state | Varchar(50) | | 10 | order\_items | order\_id | Varchar(50) | | 11 | order\_items | order\_item\_id | Varchar(50) | | 12 | order\_items | product\_id | Varchar(50) | | 13 | order\_items | seller\_id | Varchar(50) | | 14 | order\_items | shipping\_limit\_date | datetime | | 15 | order\_items | price | Float8 | | 16 | order\_items | freight\_value | Float8 | | 17 | geolocations | geolocation\_zip\_code\_prefix | int | | 18 | geolocations | geolocation\_lat | Float8 | | 19 | geolocations | geolocation\_lng | Float8 | | 20 | geolocations | geolocation\_city | Varchar(50) | | 21 | geolocations | geolocation\_state | Varchar(50) | | 22 | payments | order\_id | Varchar(50) | | 23 | payments | payment\_sequential | Int | | 24 | payments | payment\_type | Varchar(50) | | 25 | payments | payment\_installments | Int | | 26 | payments | payment\_value | Float8 | | 27 | orders | order\_id | Varchar(50) | | 28 | orders | customer\_id | Varchar(50) | | 29 | orders | order\_status | Varchar(50) | | 30 | orders | order\_purchase\_timestamp | Datetime | | 31 | orders | order\_delivered\_carrier\_date | Datetime | | 32 | orders | order\_delivered\_customer\_date | Datetime | | 33 | orders | order\_estimated\_delivery\_date | Datetime | | 34 | reviews | review\_id | Varchar(50) | | 35 | reviews | order\_id | Varchar(50) | | 36 | reviews | review\_score | Int | | 37 | reviews | review\_comment\_title | Varchar(50) | | 38 | reviews | review\_comment\_message | Varchar(50) | | 39 | reviews | review\_creation\_date | Datetime | | 40 | reviews | review\_answer\_timestamp | Datetime | | 41 | products | product\_id | Varchar(50) | | 42 | products | product\_category\_name | Varchar(50) | | 43 | products | product\_name\_length | int | | 44 | products | product\_description\_length | int | | 45 | products | product\_photos\_qty | int | | 46 | products | product\_weight\_g | int | | 47 | products | product\_length\_cm | int | | 48 | products | product\_height\_cm | int | | 49 | products | product\_width\_cm | int |  |  |  | | --- | --- | |  |  | |  |

* 1. ***Time period for which the data is given***

The data is for a period:

2016-09-04 to 2018-11-12

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

CODE:

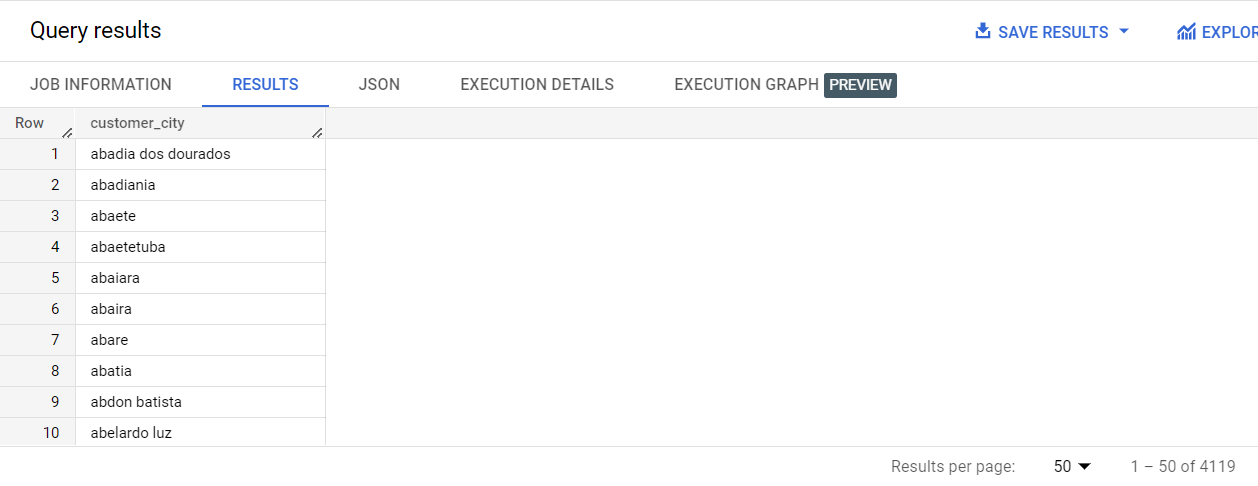
SELECT DISTINCT

   customer\_city

FROM target.customers

ORDER BY customer\_city ASC

Result up to 10 rows:



CODE:

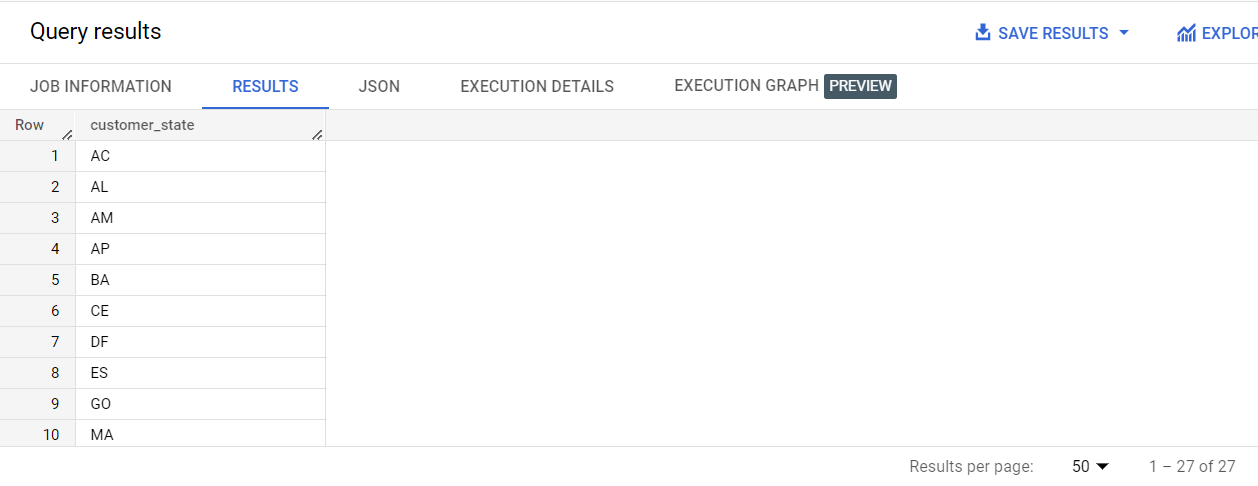
SELECT DISTINCT

   customer\_state

FROM target.customers

ORDER BY customer\_state ASC

Result up to 10 rows:



1. **In-depth Exploration:**
2. ***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?***

CODE:

SELECT DISTINCT

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

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

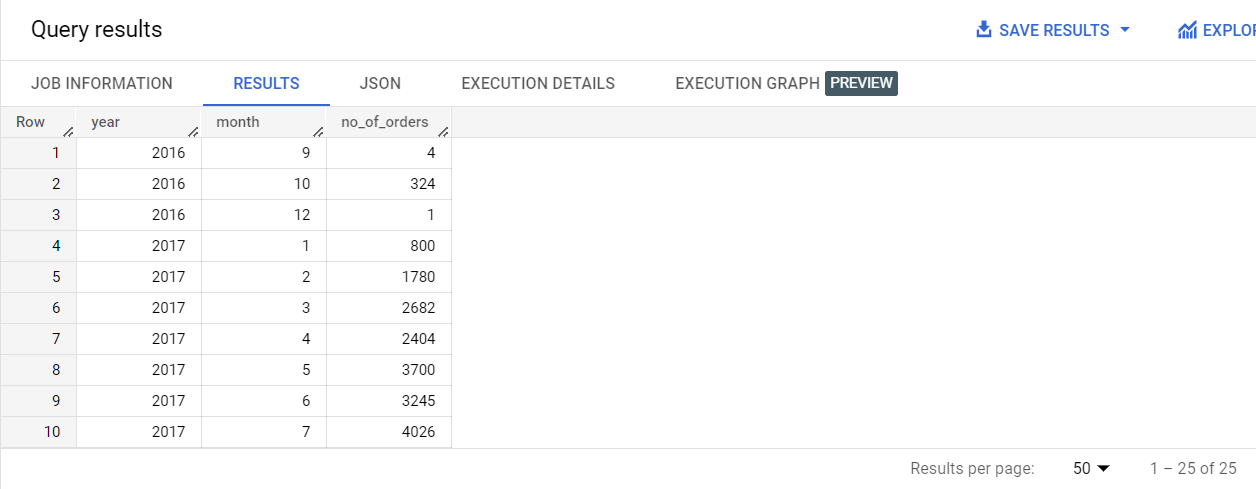
   COUNT(\*) as no\_of\_orders

FROM target.orders o

GROUP BY year,month

ORDER BY year,month

Result up to 10 rows:



Insight/Observation:

As can be seen from the data, we do see a growing trend of ecommerce in Brazil which peaks around mid-year months, i.e., 4th to 8th month. This could be attributed to several cultural and social factors. In Brazil, the period from April to August marks the fall and winter season, and during this time, people tend to spend more time at home and indoors. This means they are more likely to engage with indoor activities like online shopping. Additionally, the months of June and July include popular Brazilian holidays such as São João and Independence Day, which may drive more online shopping activity.

Recommendation:

To capitalize on this surge in sales, Target could focus on creating a strong social media strategy during this period. They could create seasonal campaigns that are focused on fall and winter products such as warm clothing, heaters, and home decor items that align with the cultural and social context of this period. For example, they could create a series of social media posts and ads featuring cozy winter outfits or home decor products that are designed to be used during the fall and winter seasons.

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

***NOTE: Assuming: Dawn – 00:00-06:00, Morning – 06:00-12:00, Afternoon – 12:00-18:00, Night - 18:00-23:59***

CODE:

SELECT

   CASE

    WHEN EXTRACT(HOUR FROM CAST(order\_purchase\_timestamp AS TIMESTAMP)) BETWEEN 0 AND 6 THEN 'Dawn'

    WHEN EXTRACT(HOUR FROM CAST(order\_purchase\_timestamp AS TIMESTAMP)) BETWEEN 6 AND 12 THEN 'Morning'

    WHEN EXTRACT(HOUR FROM CAST(order\_purchase\_timestamp AS TIMESTAMP)) BETWEEN 12 AND 18 THEN 'Afternoon'

    WHEN EXTRACT(HOUR FROM CAST(order\_purchase\_timestamp AS TIMESTAMP)) BETWEEN 18 AND 23 THEN 'Night'

  END AS purchase\_time,

   COUNT(\*) AS total\_purchases

FROM

   target.orders o

   JOIN target.customers c

   ON o.customer\_id = c.customer\_id

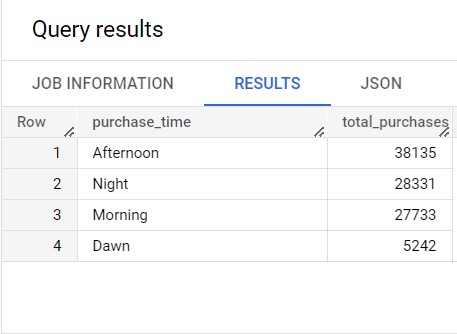
GROUP BY

   purchase\_time

ORDER BY

   total\_purchases DESC;

Result up to 10 rows:



Insight/Observation:

From the data it can be observed that the Brazilians tend to shop more during the afternoon period with the night time being a far 2nd and Morning a closer 3rd.

Recommendation:

The majority of customers are buying during the afternoon and night, so Target should optimize its marketing campaigns accordingly, targeting its advertising efforts during the afternoon and night, which are the peak times for shopping. This can include running targeted social media campaigns and offering promotions during these times.

1. **Evolution of E-commerce orders in the Brazil region:**
2. ***Get month on month orders by states***

CODE:

SELECT DISTINCT

  c.customer\_state,

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

  COUNT(\*) AS count\_

FROM target.orders o

JOIN target.customers c

ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state,month

ORDER BY c.customer\_state,month

Result up to 10 rows:



Insight/Observation:

Observation of data suggests that the sale across all states surges during the mid-year months, i.e., 4th to 8th months.

Recommendation:

They could create a series of social media posts and ads featuring cozy winter outfits or home decor products that are designed to be used during the fall and winter seasons. Additionally, they could leverage the Brazilian holidays in their social media campaigns, offering exclusive discounts and promotions to customers during these periods.

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

CODE:

SELECT

  customer\_state,

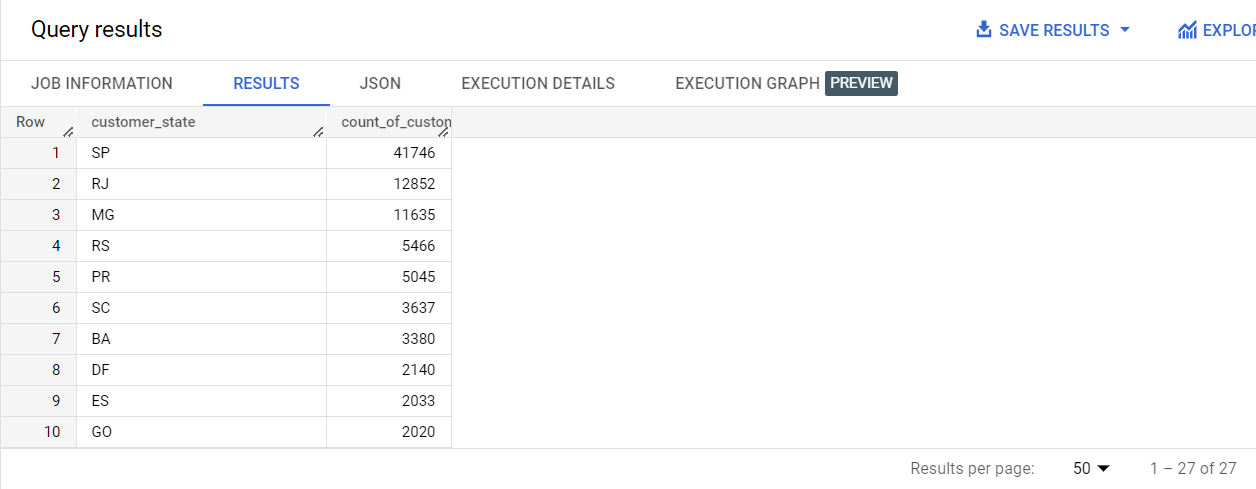
  COUNT(\*) AS count\_of\_customers

FROM target.customers

GROUP BY customer\_state

ORDER BY count\_of\_customers DESC

Result up to 10 rows:



Insight/Observation:

Here, it can be seen that the top 3 states with the highest number of sales are São Paulo (SP), Rio de Janeiro (RJ), and Minas Gerais (MG), with SP having the highest number of sales by a large margin. This might be because of the fact that SP is the most populous state in Brazil and is home to many large cities and urban areas., leading to higher concentration of potential customers.

Recommendation:

To design an appropriate growth engine for further business expansion in Brazil, Target can focus on building brand awareness and loyalty in the top 3 states with the highest sales, especially São Paulo. This could involve targeted advertising and promotions on social media platforms popular in Brazil leveraging appropriate performance marketing tools, such as Facebook and Instagram ads, as well as partnering with local influencers and bloggers to showcase their products.

1. **Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.**
2. ***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***

CODE:

SELECT

ROUND(((SUM(CASE WHEN EXTRACT(YEAR FROM CAST(o.order\_purchase\_timestamp AS DATE)) = 2018 AND EXTRACT(MONTH FROM CAST(o.order\_purchase\_timestamp AS DATE)) BETWEEN 1 AND 8 THEN p.payment\_value ELSE 0 END)

- SUM(CASE WHEN EXTRACT(YEAR FROM CAST(o.order\_purchase\_timestamp AS DATE)) = 2017 AND EXTRACT(MONTH FROM CAST(o.order\_purchase\_timestamp AS DATE)) BETWEEN 1 AND 8 THEN p.payment\_value ELSE 0 END))

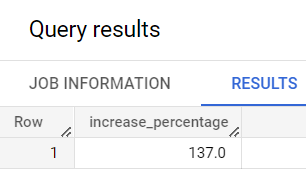
/ (SUM(CASE WHEN EXTRACT(YEAR FROM CAST(o.order\_purchase\_timestamp AS DATE)) = 2017 AND EXTRACT(MONTH FROM CAST(o.order\_purchase\_timestamp AS DATE)) BETWEEN 1 AND 8 THEN p.payment\_value ELSE 0 END))),2) \* 100 AS increase\_percentage

FROM target.orders o

JOIN target.payments p

ON o.order\_id = p.order\_id

Result up to 10 rows:



Insight/Observation:

We can observe that there is a 137% increase in total sales for the Target Ecommerce in Brazil from FY17 to FY18. (For the data pertaining to Jan to Aug months). This seems to be a significant growth given its only the second year of Target in the country. This indicates that Target has been successful in acquiring weighty market share in the early days.

Recommendation:

Target should continue to focus on better customer experience because down the road retaining the customers is going to be more important than just keep gaining new ones. Hence, a push for a better loyalty program may be a good strategy.

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

CODE:

SELECT

   c.customer\_state,

   ROUND(AVG(oi.price),2) AS avg\_price,

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

   ROUND(SUM(oi.price),2) AS total\_price,

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

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

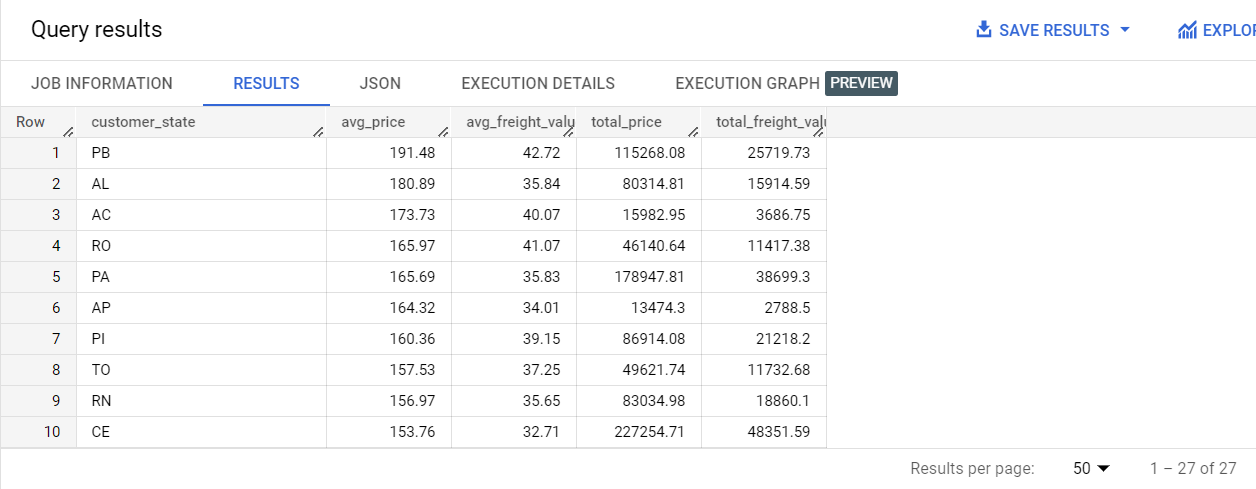
JOIN target.order\_items oi

ON oi.order\_id = o.order\_id

GROUP BY customer\_state

ORDER BY avg\_price DESC

Result up to 10 rows:



CODE:

SELECT

   c.customer\_state,

   ROUND(AVG(oi.price),2) AS avg\_price,

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

   ROUND(SUM(oi.price),2) AS total\_price,

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

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

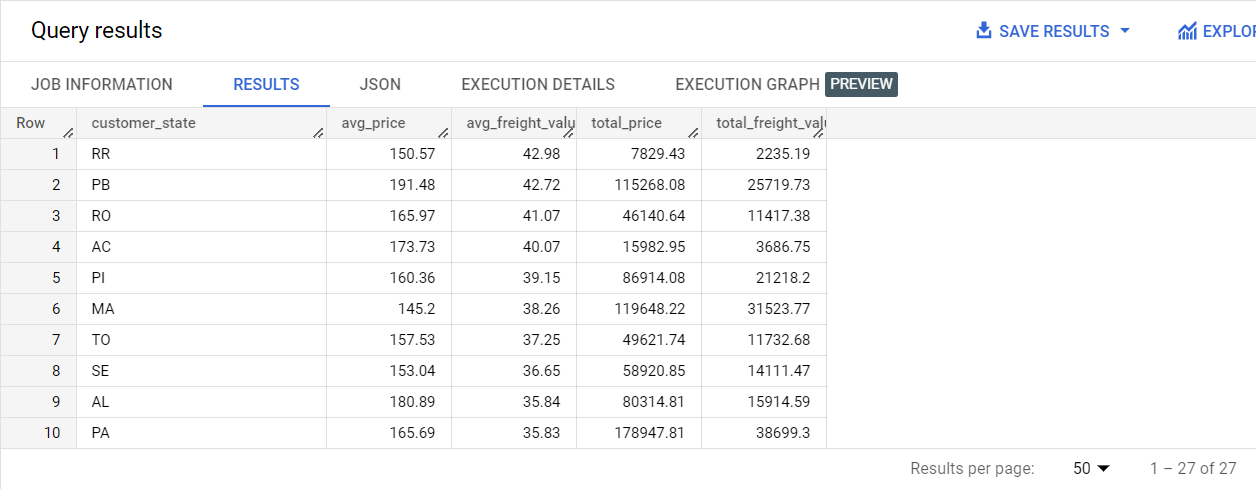
JOIN target.order\_items oi

ON oi.order\_id = o.order\_id

GROUP BY customer\_state

ORDER BY avg\_freight\_value DESC

Result up to 10 rows:



CODE:

SELECT

   c.customer\_state,

   ROUND(AVG(oi.price),2) AS avg\_price,

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

   ROUND(SUM(oi.price),2) AS total\_price,

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

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

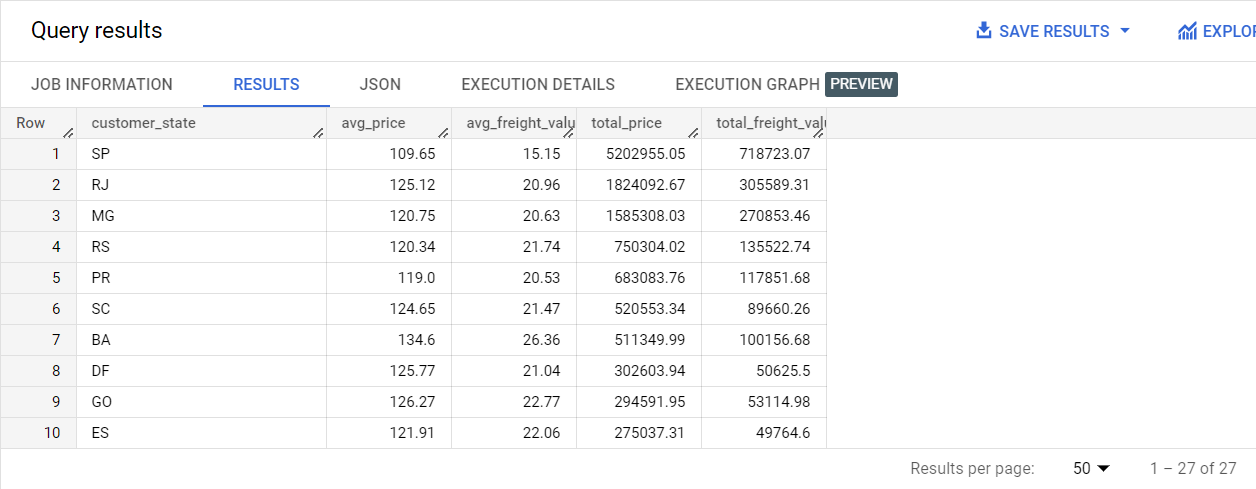
JOIN target.order\_items oi

ON oi.order\_id = o.order\_id

GROUP BY customer\_state

ORDER BY total\_price DESC

Result up to 10 rows:



CODE:

SELECT

   c.customer\_state,

   ROUND(AVG(oi.price),2) AS avg\_price,

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

   ROUND(SUM(oi.price),2) AS total\_price,

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

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

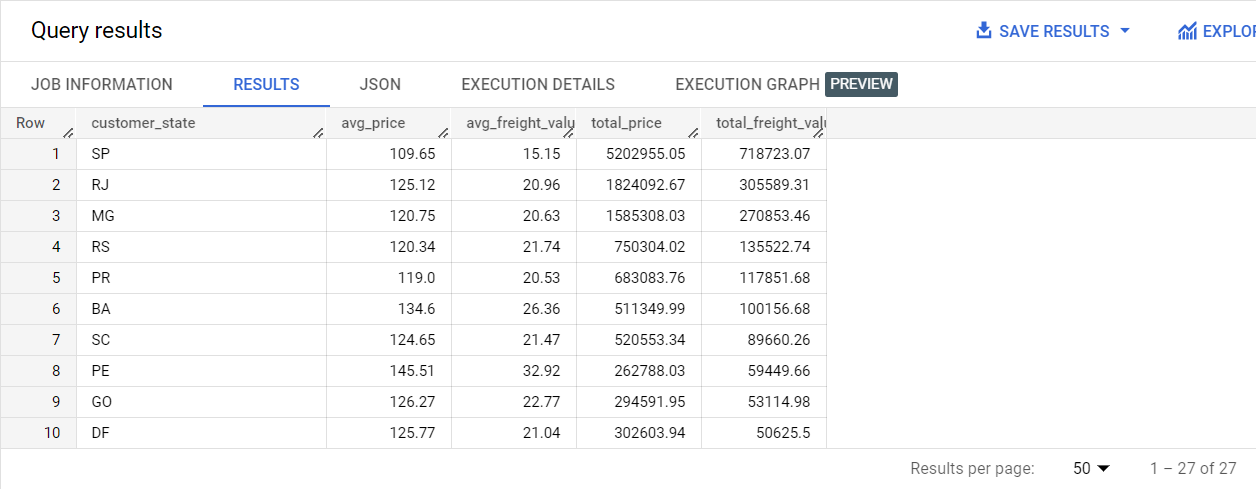
JOIN target.order\_items oi

ON oi.order\_id = o.order\_id

GROUP BY customer\_state

ORDER BY total\_freight\_value DESC

Result up to 10 rows:



Insight/Observation:

We know when it comes to sheer volume of sales, SP, RJ and MG are the states that top the chart. While, PB, AL and AC seem to be getting more higher ticket orders which suggests a financially well-off customer base in those regions.

Recommendation:

It would be recommended that Target further investigates more about the states like SP, RJ and MG through conducting customer surveys, analysing the competition, and identifying potential opportunities for expansion and try and find a duplicate audience in other states as well

1. **Analysis on sales, freight and delivery time:**
2. ***Calculate days between purchasing, delivering and estimated delivery***

CODE:

SELECT

order\_id,

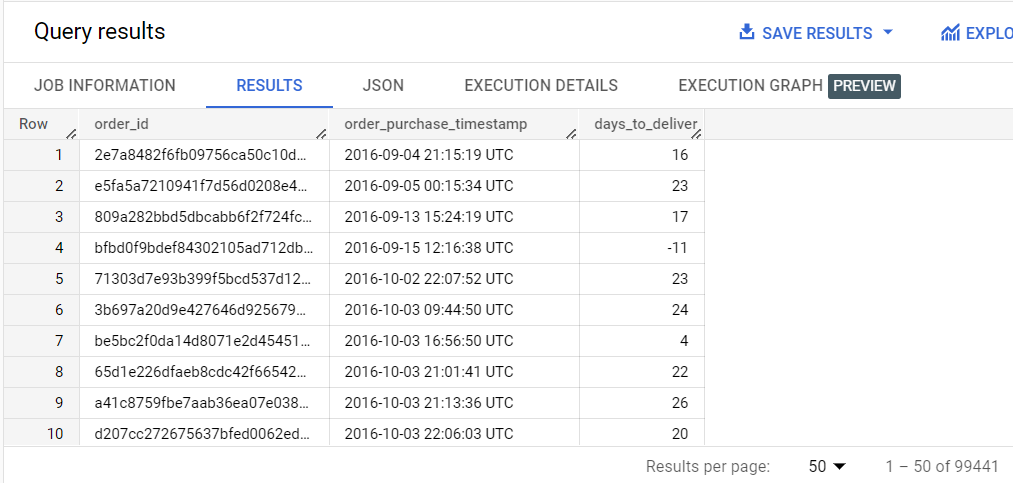
   order\_purchase\_timestamp,

   EXTRACT(DAY FROM order\_estimated\_delivery\_date) -  EXTRACT(DAY FROM order\_purchase\_timestamp) AS days\_to\_deliver

FROM target.orders

ORDER BY order\_purchase\_timestamp

Result up to 10 rows:



1. ***Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:***
   1. ***time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date***
   2. ***diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date***

CODE:

SELECT

   order\_id,

   order\_purchase\_timestamp,

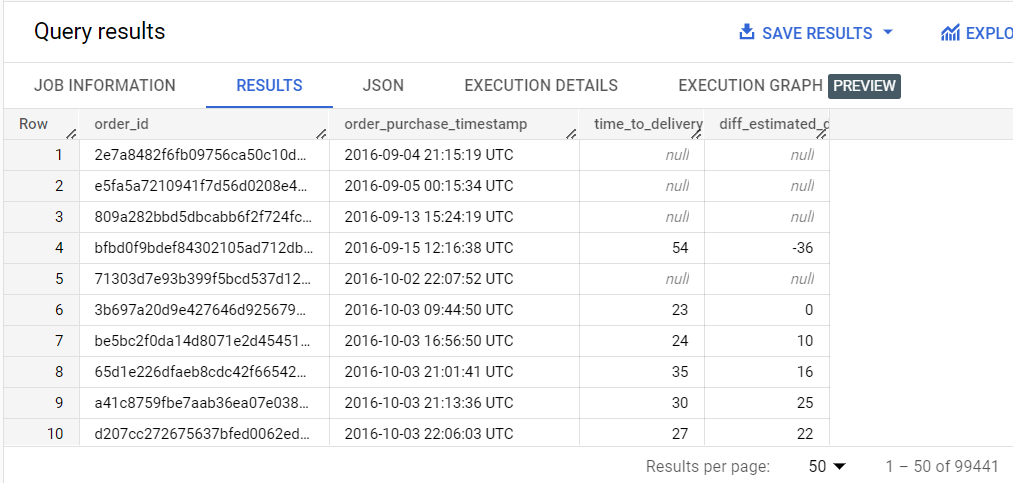
   TIMESTAMP\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) AS time\_to\_delivery,

   TIMESTAMP\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, DAY) AS diff\_estimated\_delivery

FROM target.orders

ORDER BY order\_purchase\_timestamp

Result up to 10 rows:



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

CODE:

SELECT

   c.customer\_state,

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

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date, DAY)),2) AS avg\_diff\_estimated\_delivery

FROM target.customers c

JOIN target.orders o

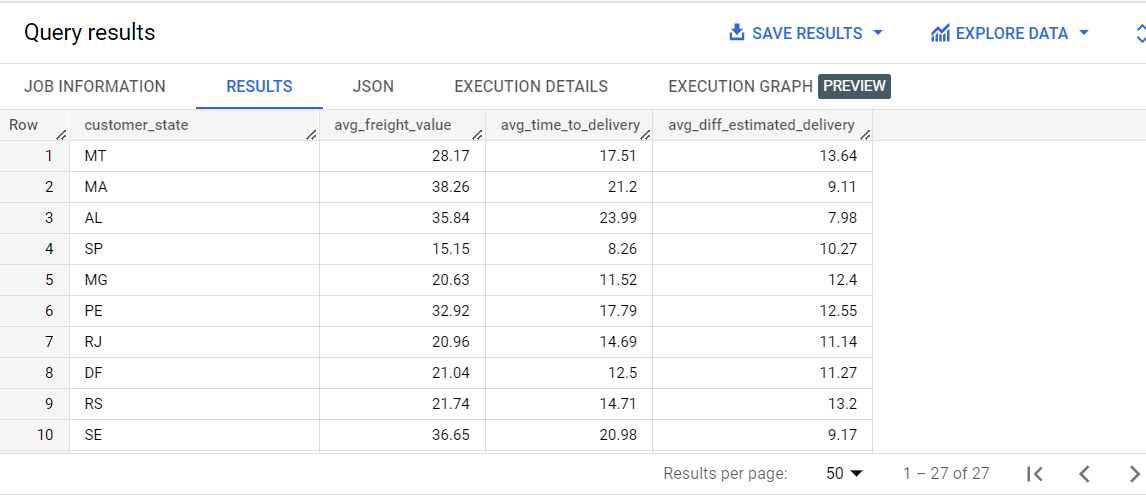
ON o.customer\_id = c.customer\_id

JOIN target.order\_items oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

Result up to 10 rows:



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

CODE:

SELECT

   c.customer\_state,

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

   ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date, DAY)),2) AS avg\_diff\_estimated\_delivery

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

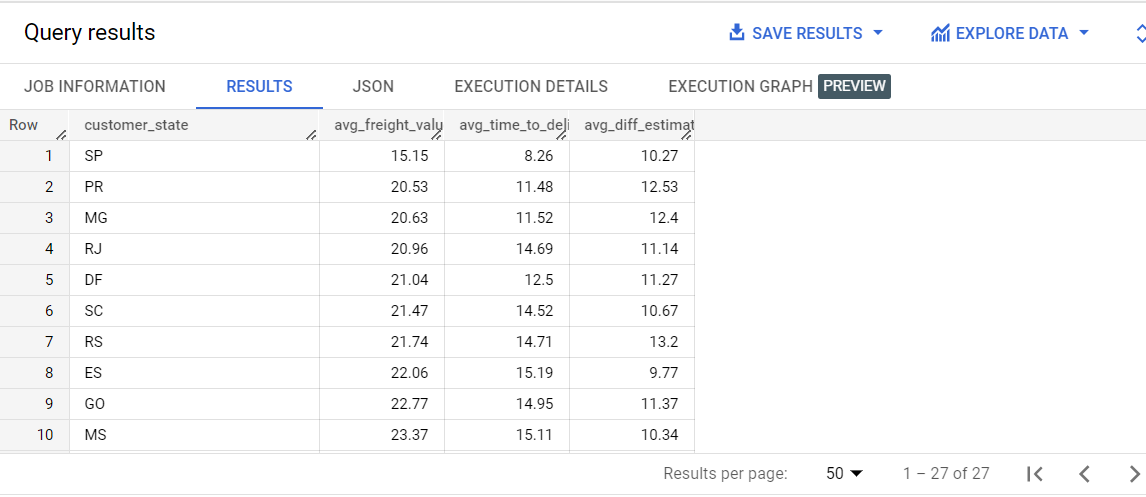
JOIN target.order\_items oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_freight\_value

Result up to 10 rows:



1. ***Top 5 states with highest average freight value - sort in desc limit 5***

CODE:

SELECT

   c.customer\_state,

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

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date, DAY)),2) AS avg\_diff\_estimated\_delivery

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

JOIN target.order\_items oi

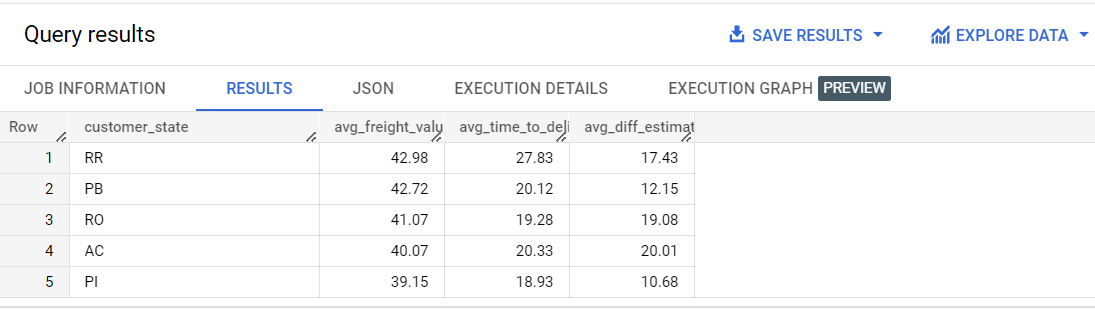
ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_freight\_value DESC

LIMIT 5

Result up to 5 rows:



1. ***Top 5 states with lowest average freight value - sort in asc limit 5***

CODE:

SELECT

   c.customer\_state,

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

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date, DAY)),2) AS avg\_diff\_estimated\_delivery

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

JOIN target.order\_items oi

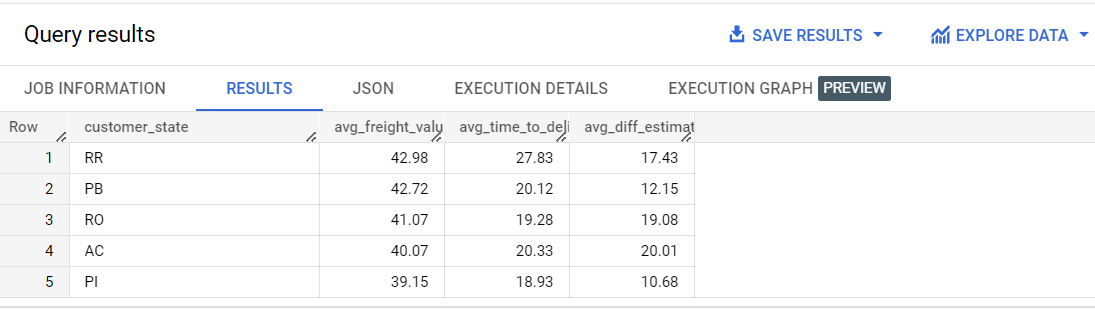
ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_freight\_value ASC

LIMIT 5

Result up to 5 rows:



1. ***Top 5 states with highest average time to delivery***

CODE:

SELECT

   c.customer\_state,

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

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date, DAY)),2) AS avg\_diff\_estimated\_delivery

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

JOIN target.order\_items oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_time\_to\_delivery DESC

LIMIT 5

Result up to 5 rows:



1. ***Top 5 states with lowest average time to delivery***

CODE:

SELECT

   c.customer\_state,

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

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date, DAY)),2) AS avg\_diff\_estimated\_delivery

FROM target.customers c

JOIN target.orders o

ON o.customer\_id = c.customer\_id

JOIN target.order\_items oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_time\_to\_delivery ASC

LIMIT 5

Result up to 5 rows:



1. ***Top 5 states where delivery is really fast compared to estimated date***

CODE:

SELECT

    c.customer\_state,

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

  AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_purchase\_timestamp, DAY)) AS avg\_diff\_estimated\_delivery,

  (AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)))/(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_purchase\_timestamp, DAY))) as ratio

FROM target.customers c

JOIN target.orders o

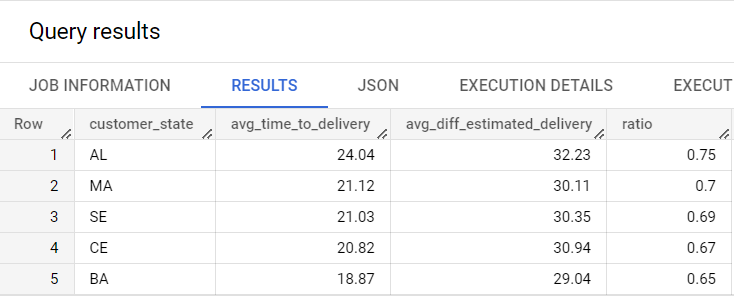
ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY ratio DESC

LIMIT 5

Result up to 10 rows:



1. ***Top 5 states where delivery is not so fast compared to estimated date***

CODE:

SELECT

   c.customer\_state,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)),2) AS avg\_time\_to\_delivery,

  ROUND(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_purchase\_timestamp, DAY)),2) AS avg\_diff\_estimated\_delivery,

  ROUND((AVG(TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,DAY)))/(AVG(TIMESTAMP\_DIFF(o.order\_estimated\_delivery\_date,o.order\_purchase\_timestamp, DAY))),2) as ratio

FROM target.customers c

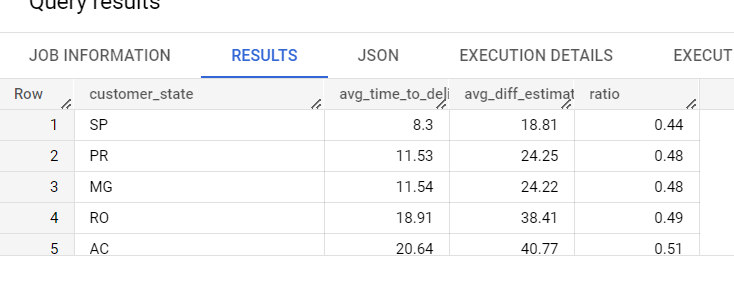
JOIN target.orders o

ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY ratio ASC

Result up to 10 rows:



1. **Payment type analysis:**
2. ***Month over Month count of orders for different payment types:***

CODE:

SELECT

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

  COUNT(CASE WHEN p.payment\_type = 'credit card' THEN 1 END) AS credit\_card,

  COUNT(CASE WHEN p.payment\_type = 'UPI' THEN 1 END) AS upi,

  COUNT(CASE WHEN p.payment\_type = 'debit card' THEN 1 END) AS debit\_card,

  COUNT(CASE WHEN p.payment\_type = 'voucher' THEN 1 END) AS voucher,

  COUNT(CASE WHEN p.payment\_type = 'not defined' THEN 1 END) AS not\_defined

FROM target.orders AS o

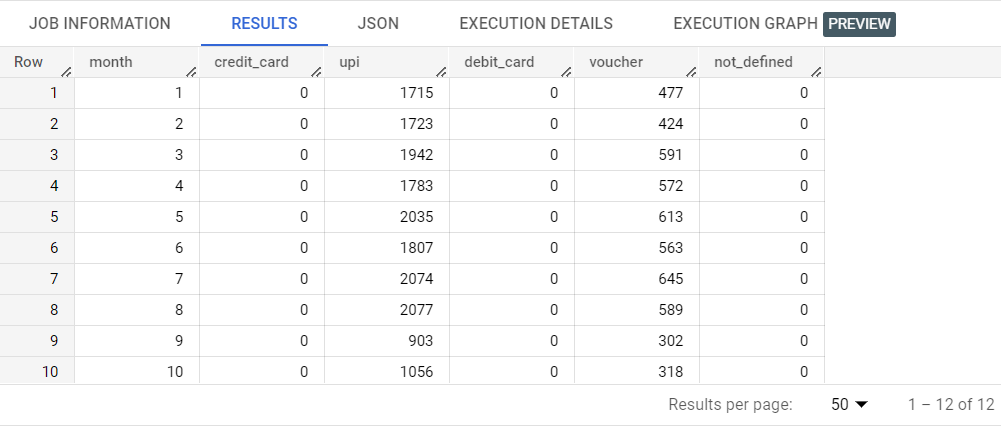
JOIN target.payments AS p

ON o.order\_id = p.order\_id

GROUP BY month

ORDER BY month

Result up to 10 rows:



Insight/Observation:

The data suggests that only UPI and Vouchers are the payment types being used by the Brazilian customers. Now, UPI is a payment type used in India so, it’s unlikely that Brazilian customers are using UPI as payment type, but if the data suggests it, this tells one of the two things about the customers. Either the Brazilians are slow at adopting to the traditional payment techniques of credit card and debit card, or they have leapfrogged the technology into using the UPI.

1. ***Count of orders based on the no. of payment instalments:***

CODE:

SELECT

   p.payment\_installments,

     COUNT(\*) AS count\_of\_orders

FROM target.orders o

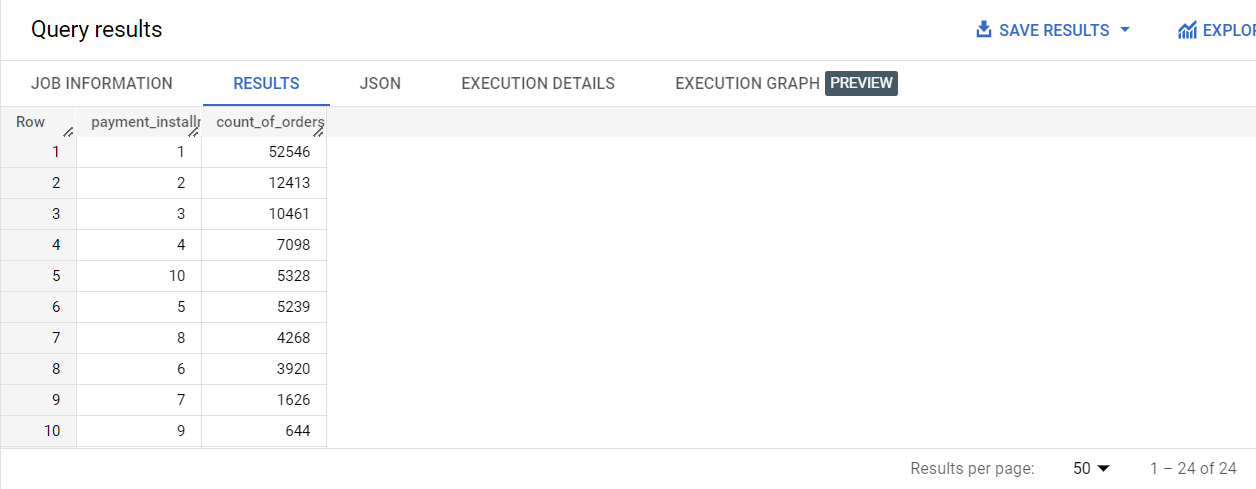
JOIN target.payments p

ON o.order\_id = p.order\_id

GROUP BY p.payment\_installments

ORDER BY count\_of\_orders DESC

Result up to 10 rows:



Insight/Observation:

Here, we can see that mostly the payments ate being done in a single instalment itself which is an indicator of a financially strong customer segment that is getting attracted towards the company’s products on its ecommerce platform. Though, we see a large number of orders being paid in 2, 3 or 4 instalments too, which indicates an audience base that prefers paying in multiple payments.

Recommendation:

Looking at the number of orders paid in multiple instalments, the company needs to accommodate, if it is not already, Buy Now Pay Later facilities by having third party collaborations. This might help increase those numbers. Also, Target can incentivise the customers to pay in a single payment by offering attractive discounts.