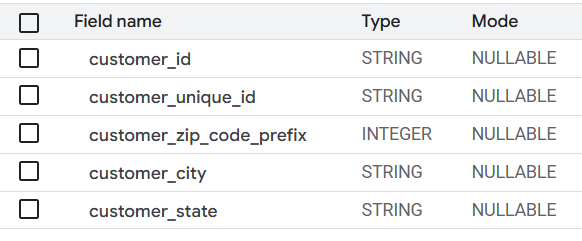
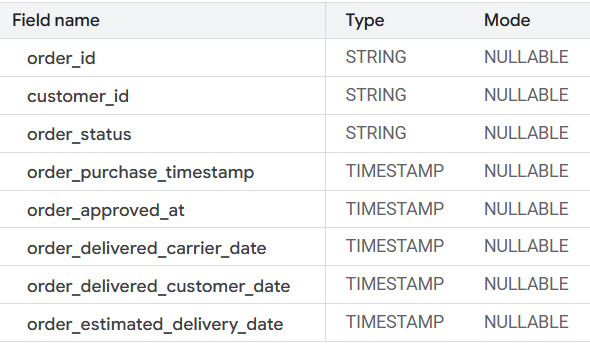
**Problem Statement:**

**Target** seeks to improve its operations and customer experience in Brazil by analysing historical order data FROM 2016 to 2018. The objective is to uncover insights related to order fulfilment, pricing, payments, shipping, customer behaviour, product performance, and reviews. These insights will help Target enhance operational efficiency, optimize pricing and delivery strategies, and improve customer satisfaction to strengthen its position in the Brazilian retail market.

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



* **Date type of all columns in the “orders” table.**



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

**Code:**

SELECT

min(order\_purchase\_timestamp) as first\_order,

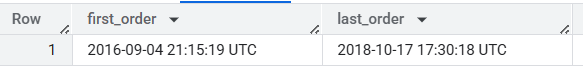
max(order\_purchase\_timestamp) as last\_order

FROM

`scalerproject-454408.target\_sql\_project.orders`

;

**Output:**



**First order was at 4th September 2016 and the Last Order was at 17th October 2018**

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

**Code:**

SELECT

COUNT(distinct c.customer\_city) as city\_COUNT,

COUNT(distinct c.customer\_state) as state\_COUNT

FROM

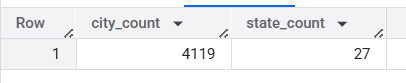
`scalerproject-454408.target\_sql\_project.customers` as c

JOIN

`scalerproject-454408.target\_sql\_project.orders` as o

on c.customer\_id = o.customer\_id ;

**Output:**



Total City COUNT is at 4119 and Total State COUNT is at 27

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

**Code:**

SELECT

EXTRACT(year FROM order\_purchase\_timestamp) as year,

EXTRACT(month FROM order\_purchase\_timestamp) as month,

COUNT(order\_id) as order\_COUNT

FROM

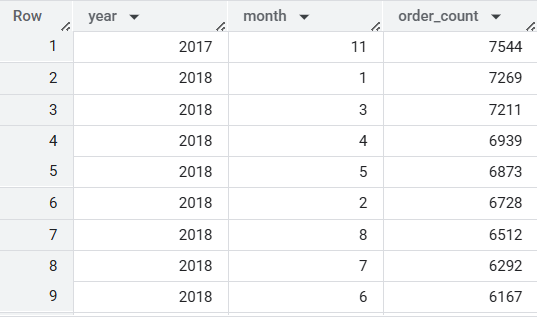
`scalerproject-454408.target\_sql\_project.orders`

GROUP BY year,month

ORDER BY order\_COUNT DESC,year,month

;

**Output:**



Highest order count of 7544 was at November of 2017 followed by Jan of 2018

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

**Code:**

SELECT

EXTRACT(month FROM order\_purchase\_timestamp) as month,

COUNT(order\_id) as order\_count

FROM

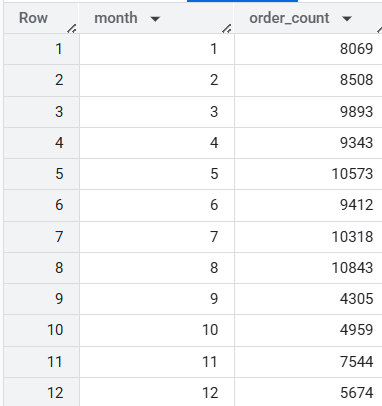
`scalerproject-454408.target\_sql\_project.orders`

GROUP BY month

ORDER BY order\_count DESC,month

;

**Output:**



**#Year wise Order Count**

**Code:**

SELECT

EXTRACT(year FROM order\_purchase\_timestamp) as year,

COUNT(order\_id) as order\_count

FROM

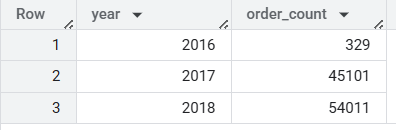
`scalerproject-454408.target\_sql\_project.orders`

GROUP BY year

ORDER BY order\_count DESC,year

;

**Output:**



**# During 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**

**Code:**

SELECT

CASE

    WHEN EXTRACT(hour FROM order\_purchase\_timestamp) between 0 and 6 then 'Dawn'

    WHEN EXTRACT(hour FROM order\_purchase\_timestamp) between 7 and 12 then 'Mornings'

    WHEN EXTRACT(hour FROM order\_purchase\_timestamp) between 13 and 18 then 'Afternoon'

    else 'Night' end as time\_of\_the\_day,

  COUNT(order\_id) as order\_ count

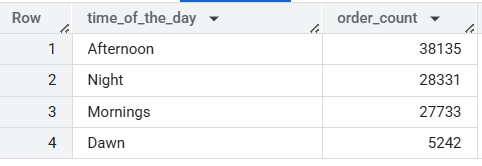
FROM

  `scalerproject-454408.target\_sql\_project.orders`

GROUP BY time\_of\_the\_day

ORDER BY order\_ count DESC ;

**Output:**



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

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

**Code:**

SELECT

c.customer\_state,

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

COUNT(o.order\_id) as no\_of\_orders

FROM

`scalerproject-454408.target\_sql\_project.customers` as c

JOIN

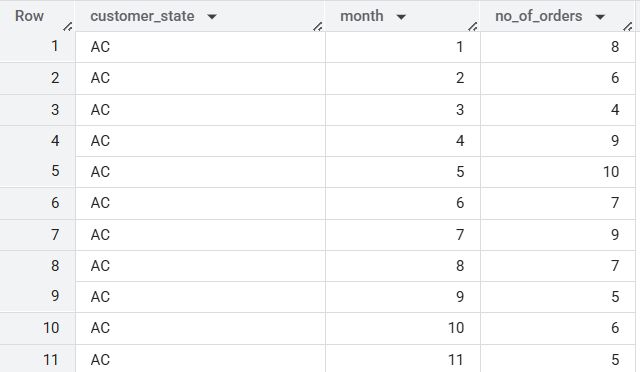
`scalerproject-454408.target\_sql\_project.orders` as o

on c.customer\_id = o.customer\_id

GROUP BY c.customer\_state, month

ORDER BY c.customer\_state, month ;

**Output:**



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

**Code:**

SELECT

customer\_state,

COUNT(customer\_id) as customer\_ count

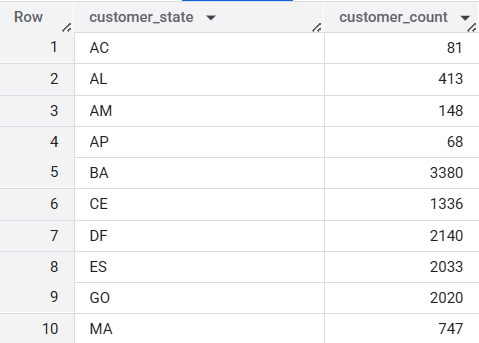
FROM

`scalerproject-454408.target\_sql\_project.customers`

GROUP BY customer\_state

ORDER BY customer\_state ;

**Output:**



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

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

**Method 1:**

**Code:**

SELECT

ROUND(sum(CASE

WHEN EXTRACT(year FROM order\_purchase\_timestamp) = 2017 then p.payment\_value

else 0 end),2) as cost\_of\_orders\_2017,

  ROUND(sum(CASE

    WHEN EXTRACT(year FROM o.order\_purchase\_timestamp) = 2018 then p.payment\_value

else 0 end),2) as cost\_of\_orders\_2018,

  (sum(CASE

WHEN EXTRACT(year FROM o.order\_purchase\_timestamp) = 2018 then p.payment\_value else 0 end) \*100 /sum(CASE

WHEN EXTRACT(year FROM o.order\_purchase\_timestamp) = 2017 then p.payment\_value else 0 end))-100 as percent\_increase

FROM

`scalerproject-454408.target\_sql\_project.orders` as o

JOIN

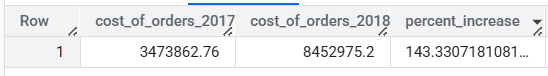
`scalerproject-454408.target\_sql\_project.payments` as p

on o.order\_id = p.order\_id

WHERE (EXTRACT(year FROM o.order\_purchase\_timestamp) between 2017 and 2018) and

(EXTRACT(month FROM o.order\_purchase\_timestamp) between 1 and 8) and o.order\_status='delivered' ;

**Output:**



**Method 2:**

**Code:**

With CLT as (

SELECT

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

  ROUND(sum(p.payment\_value),2) as total\_value

FROM

`scalerproject-454408.target\_sql\_project.orders` as o

JOIN

`scalerproject-454408.target\_sql\_project.payments` as p

on o.order\_id = p.order\_id

WHERE (EXTRACT(year FROM o.order\_purchase\_timestamp) between 2017 and 2018) and

(EXTRACT(month FROM o.order\_purchase\_timestamp) between 1 and 8) and o.order\_status='delivered'

GROUP BY year

)

SELECT

year,

  ROUND(total\_value,2) as total\_value,

  Lag(total\_value) over(ORDER BY year ASC ) as prev\_year,

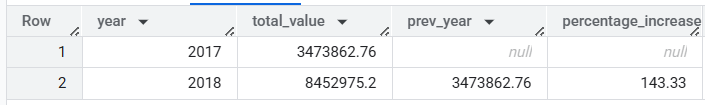
ROUND((total\_value \*100/(lag(total\_value) over(ORDER BY year ASC )))-100,2) as percentage\_increase

FROM CLT

ORDER BY total\_value ASC

;

**Output:**



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

**Code:**

SELECT

c.customer\_state,

ROUND(avg(p.payment\_value),2) as Average\_value,

ROUND(sum(p.payment\_value),2) as Total\_value

FROM

`scalerproject-454408.target\_sql\_project.orders` as o

JOIN

`scalerproject-454408.target\_sql\_project.payments` as p

on p.order\_id = o.order\_id

JOIN

`scalerproject-454408.target\_sql\_project.customers` as c

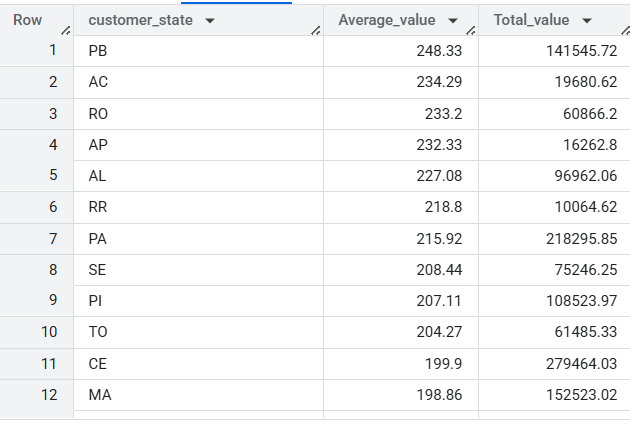
on c.customer\_id = o.customer\_id

GROUP BY c.customer\_state

ORDER BY Average\_value DESC, Total\_value DESC

;

**Output:**



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

**Code:**

SELECT

order\_id,

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

DATE\_DIFF(order\_delivered\_customer\_date,order\_estimated\_delivery\_date,DAY) as diff\_estimated\_delivery

FROM

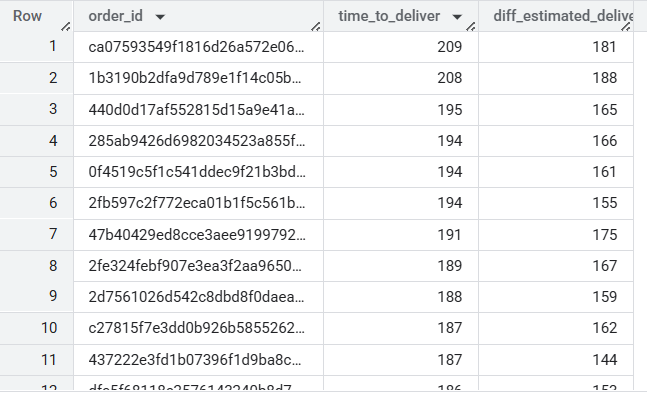
`scalerproject-454408.target\_sql\_project.orders`

WHERE order\_status = 'delivered'

ORDER BY time\_to\_deliver DESC, diff\_estimated\_delivery DESC

;

**Output:**



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

**# Highest Avg Freight Value:**

**Code:**

SELECT

c.customer\_state,

  ROUND(avg(ot.freight\_value),2) as Average\_freight\_value

FROM

`scalerproject-454408.target\_sql\_project.customers` as c

JOIN

`scalerproject-454408.target\_sql\_project.orders` as o

on c.customer\_id = o.customer\_id

JOIN

`scalerproject-454408.target\_sql\_project.order\_items` as ot

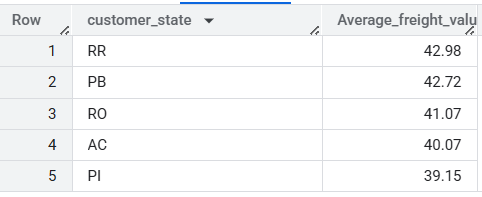
on ot.order\_id = o.order\_id

GROUP BY c.customer\_state

ORDER BY Average\_freight\_value DESC

LIMIT 5 ;

**Output:**



**# Lowest Average Freight value:**

**Code:**

SELECT

c.customer\_state,

  ROUND(avg(ot.freight\_value),2) as Average\_freight\_value

FROM

`scalerproject-454408.target\_sql\_project.customers` as c

JOIN

`scalerproject-454408.target\_sql\_project.orders` as o

on c.customer\_id = o.customer\_id

JOIN

`scalerproject-454408.target\_sql\_project.order\_items` as ot

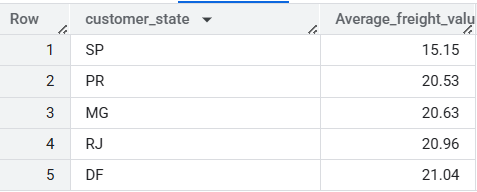
on ot.order\_id = o.order\_id

GROUP BY c.customer\_state

ORDER BY Average\_freight\_value ASC

LIMIT 5 ;

**Output:**



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

**# top 5 states with Lowest Average Delivery time**

**Code:**

SELECT

c.customer\_state, ROUND(avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, DAY)),2) as delivery\_time

FROM

`scalerproject-454408.target\_sql\_project.customers` as c

JOIN

`scalerproject-454408.target\_sql\_project.orders` as o

on c.customer\_id = o.customer\_id

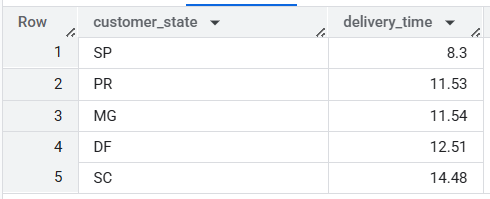
GROUP BY c.customer\_state

ORDER BY delivery\_time ASC

LIMIT 5

;

**Output:**



**# Top 5 states with Highest Average Delivery Time :**

**Code:**

SELECT

  c.customer\_state,

  ROUND(avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, DAY)),2) as delivery\_time

FROM

`scalerproject-454408.target\_sql\_project.customers` as c

JOIN

`scalerproject-454408.target\_sql\_project.orders` as o

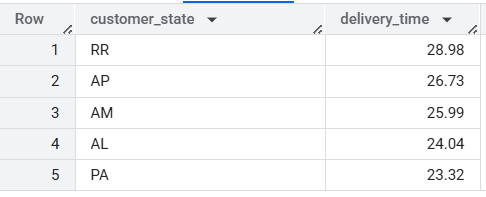
on c.customer\_id = o.customer\_id

GROUP BY c.customer\_state

ORDER BY delivery\_time DESC

LIMIT 5 ;

**Output:**



**# Find out the top 5 states WHERE the order delivery is really fast as compared to the estimated date of delivery. Difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.**

**Code:**

with delivery\_time as (

SELECT

c.customer\_state,

c.customer\_id,

o.order\_id,

DATE\_DIFF(o.order\_delivered\_customer\_date,o.order\_estimated\_delivery\_date,DAY) as delivery\_time\_diff

FROM

`scalerproject-454408.target\_sql\_project.orders` as o

JOIN

`scalerproject-454408.target\_sql\_project.customers` as c

on o.customer\_id = c.customer\_id

WHERE o.order\_status = 'delivered'

)

SELECT

customer\_state,

avg(delivery\_time\_diff) as aver

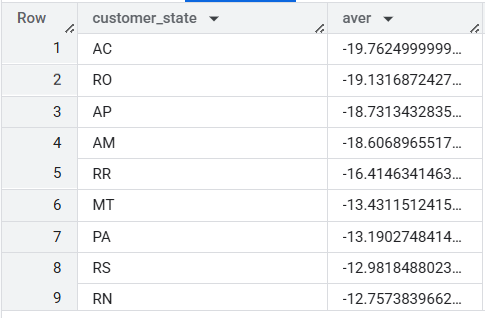
FROM

delivery\_time

GROUP BY customer\_state

ORDER BY aver ASC ;

**Output:**



**# Analysis based on the payments:**

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

**Code:**

SELECT

EXTRACT(month FROM o.order\_purchase\_timestamp) as months,

p.payment\_type,

COUNT(o.order\_id) as no\_of\_orders

FROM

`scalerproject-454408.target\_sql\_project.orders` as o

JOIN

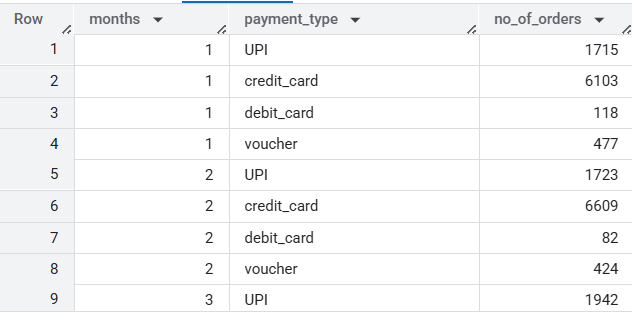
`scalerproject-454408.target\_sql\_project.payments` as p

on o.order\_id = p.order\_id

GROUP BY months, payment\_type

ORDER BY months, payment\_type, no\_of\_orders ASC ;

**Output:**



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

**Code:**

SELECT

p.payment\_installments,

COUNT(o.order\_id) as no\_of\_orders\_placed

FROM

`scalerproject-454408.target\_sql\_project.payments` as p

JOIN

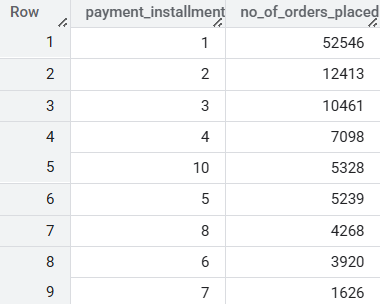
`scalerproject-454408.target\_sql\_project.orders` as o

on p.order\_id = o.order\_id

GROUP BY p.payment\_installments

ORDER BY no\_of\_orders\_placed DESC ;

**Output:**



**Data Analysis Insights:**

* First order was at 4th September 2016 and the Last Order was at 17th October 2018
* Order details of Customers from total of 4119 city from 27 states
* Highest order count was at November 2017 followed by January to May of 2018
* Overall, August, May and July month contributed highest order count whereas September and October contributed less order count
* Though we had order details till October 2018 still 2018 contributed highest order counts of 54,011 compare to 2017 with 45,101
* when we compare day-part, a greater number of orders were placed at Afternoon (13 to 18hrs) followed by Night and Dawn (0 to 6hrs) has the lowest order count
* When we compare state wise order month on month, state='SP' contributed highest number of orders on August, May and July followed by other states
* State wise customer count, State-'SP' has highest customer count which actually matches with previous observation that 'SP' had the highest order count, RJ and MG were the second and third highest in customer count
* There is increase of 143% in Order value at 2018 compare to 2017 which matches with the last observation that 2018 has highest order count compare to 2017
* while observing Payment value it’s clear that 'PB' has the highest average payment value followed by AC and RO state. In terms of Total value state-'CE' is the highest
* RP, PB, RO were the top 3 in highest Freight value
* SP, PR, MG were the top 3 in lowest Freight value
* SP has the lowest delivery time of 8.3 which in return reflects in highest sales and order count followed by PR with 11.53 and MG with 11.54
* PR, AP and AM have the highest delivery time of above 26
* In states AC, RO, AP, AM and RR actual delivery date is faster than the estimated delivery date
* Highest number of orders were done through credit card followed by UPI and then vouchers and Debit card
* Top 3 payment installments basic of order counts are 1,2,3 with total of 75k orders

placed and lowest was 22 and 23 with only one order

**Recommendations for Target:**

**Sales and Region Focus:**

* Focus sales and promotions in high-performing states: SP, RJ, MG
* Expand campaigns in mid-tier but high-value states: CE, PB, RO
* Leverage customer density in SP to test new products and loyalty programs

**Logistics and Delivery Optimization**

* Reduce delivery times in slow regions: PR, AP, AM (over 26 days)
* Update estimated delivery dates where actual delivery is faster: AC, RO, AP, AM, RR
* Replicate SP’s logistics model (8.3-day average) in other regions for improved efficiency

**Payment Method Strategy:**

* Promote Credit Card and UPI options — top 2 payment methods
* Focus on 1–3 installments options, which account for ~75% of orders
* Avoid complex installments options (20+), as they have minimal adoption

**Time & Season-Based Marketing**

* Run sales and campaigns during peak ordering times: Afternoon (13–18 hrs)
* Maximize campaigns in high-order months: August, May, July
* Plan November promotions, leveraging past success (e.g., Black Friday bump)

**Order Value Optimization**

* Explore high average payment states: PB, AC, RO for premium offerings
* Offer premium bundles or loyalty perks targeting high-value regions

**Inventory & Demand Planning**

* Align inventory forecasting with 2018’s 143% YoY order value growth
* Prioritize stock placement in high-demand states: SP, RJ, MG

**Localized Operational Strategy**

* Develop state-specific dashboards for delivery, payments, and order trends
* Assign regional leads to tailor strategies based on local performance