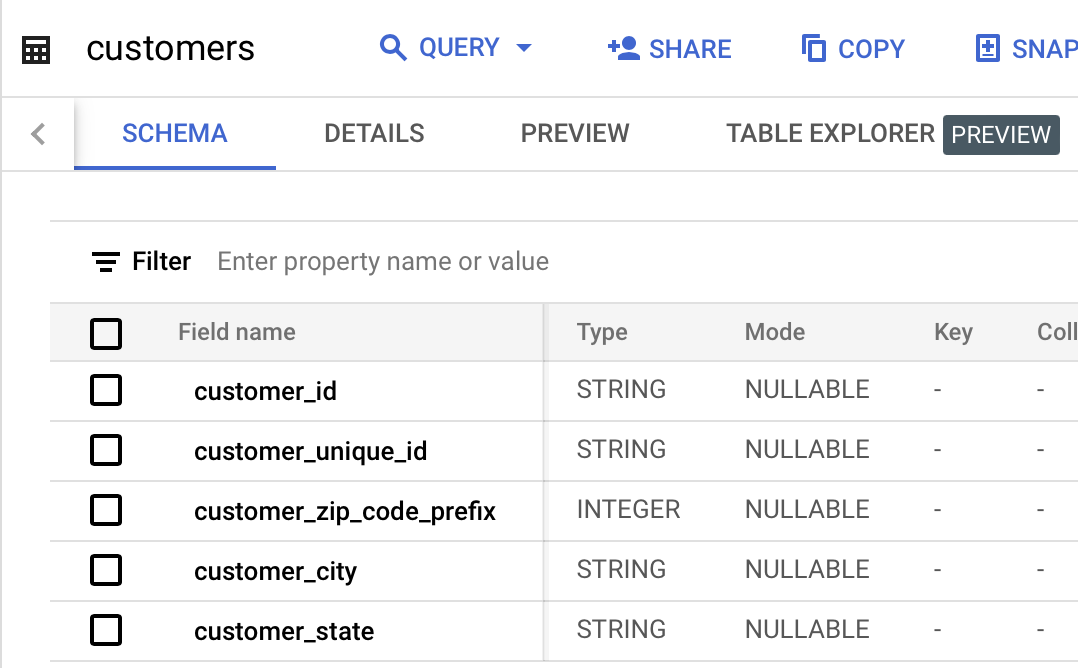
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
   1. Data type of all columns in the "customers" table.



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

select customer\_id, order\_purchase\_timestamp

from `target\_sql.orders` where order\_purchase\_timestamp in (select min(order\_purchase\_timestamp)

from `target\_sql.orders`)

| Row | customer\_id | first\_order\_date |  |
| --- | --- | --- | --- |
| 1 | 08c5351a6aca1c1589a38f244edeee9d | 2016-09-04 21:15:19 UTC |  |

select customer\_id, order\_purchase\_timestamp as last\_order\_date

from `target\_sql.orders` where order\_purchase\_timestamp in (select max(order\_purchase\_timestamp)

from `target\_sql.orders`)

| Row | customer\_id | last\_order\_date |  |
| --- | --- | --- | --- |
| 1 | a4b417188addbc05b26b72d5e44837a1 | * + 1. 7:30:18 UTC |  |

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

select

count(distinct customer\_city) as number\_of\_cities,

count(distinct customer\_state) as number\_of\_states

from `long-leaf-433203-j5.target\_sql.customers`

| Row | number\_of\_cities | number\_of\_states |  |
| --- | --- | --- | --- |
| 1 | 4119 | 27 |  |

**2. In-depth Exploration:**

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

|  |  |  |
| --- | --- | --- |
| **month** | **year** | **month\_wise\_order\_count** |
| **9** | 2016 | 4 |
| **10** | 2016 | 324 |
| **12** | 2016 | 1 |
| **1** | 2017 | 800 |
| **2** | 2017 | 1780 |
| **3** | 2017 | 2682 |
| **4** | 2017 | 2404 |
| **5** | 2017 | 3700 |
| **6** | 2017 | 3245 |
| **7** | 2017 | 4026 |
| **8** | 2017 | 4331 |
| **9** | 2017 | 4285 |
| **10** | 2017 | 4631 |
| **11** | 2017 | 7544 |
| **12** | 2017 | 5673 |
| **1** | 2018 | 7269 |
| **2** | 2018 | 6728 |
| **3** | 2018 | 7211 |
| **4** | 2018 | 6939 |
| **5** | 2018 | 6873 |
| **6** | 2018 | 6167 |
| **7** | 2018 | 6292 |
| **8** | 2018 | 6512 |
| **9** | 2018 | 16 |
| **10** | 2018 | 4 |

As, you can see from the above highlighted sections, where shows increase trend in month wise order count.

This is the query –

select extract(month from order\_purchase\_timestamp) month,

extract(year from order\_purchase\_timestamp) year,

count(\*) as month\_wise\_orders\_count

from `long-leaf-433203-j5.target\_sql.orders`

group by 1,2

order by 2,1,3 asc

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

|  |  |  |  |
| --- | --- | --- | --- |
| Row | Year | season | monthly\_seasonality |
| 1 | 2016 | Winter | 1 |
| 2 | 2016 | Fall | 328 |
| 3 | 2017 | Winter | 8253 |
| 4 | 2017 | Spring | 8786 |
| 5 | 2017 | Summer | 11602 |
| 6 | 2017 | Fall | 16460 |
| 7 | 2018 | Fall | 20 |
| 8 | 2018 | Winter | 13997 |
| 9 | 2018 | Summer | 18971 |
| 10 | 2018 | Spring | 21023 |

Select

extract( year from order\_purchase\_timestamp) as Year,

case when extract( month from order\_purchase\_timestamp) in (12,1,2) then 'Winter'

when extract( month from order\_purchase\_timestamp) in (3,4,5) then 'Spring'

when extract( month from order\_purchase\_timestamp) in (6,7,8) then 'Summer'

else 'Fall'end as season,

count(\*) as monthly\_seasonality

from `long-leaf-433203-j5.target\_sql.orders`

group by 1,2

order by 1,3 asc

* 1. 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

Select

case when extract( hour from order\_purchase\_timestamp) in (0,1,2,3,4,5,6) then 'Dawn'

when extract( hour from order\_purchase\_timestamp) in (7,8,9,10,11,12) then 'Mornings'

when extract( hour from order\_purchase\_timestamp) in (13,14,15,16,17,18) then 'Afternoon'

else 'Night'end as Hour,

count(\*) as orders\_count

from `long-leaf-433203-j5.target\_sql.orders`

group by 1

order by 1,2 desc

|  |  |  |  |
| --- | --- | --- | --- |
| Row | Hour | orders\_count |  |
| 1 | Afternoon | 38135 |  |
| 2 | Dawn | 5242 |  |
| 3 | Mornings | 27733 |  |
| 4 | Night | 28331 |  |

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Row | yr\_mon | state | monthly\_orders |  |
| 1 | 2016-09 | RR | 1 |  |
| 2 | 2016-09 | RS | 1 |  |
| 3 | 2016-09 | SP | 2 |  |
| 4 | 2016-10 | RR | 1 |  |
| 5 | 2016-10 | PB | 1 |  |
| 6 | 2016-10 | PI | 1 |  |
| 7 | 2016-10 | AL | 2 |  |
| 8 | 2016-10 | MT | 3 |  |

Select

format\_date('%Y-%m', order\_purchase\_timestamp) as yr\_mon,

--extract( month from order\_purchase\_timestamp) as month, (can use this also, but this doesn't make much sense as year is not there)

c.customer\_state as state,

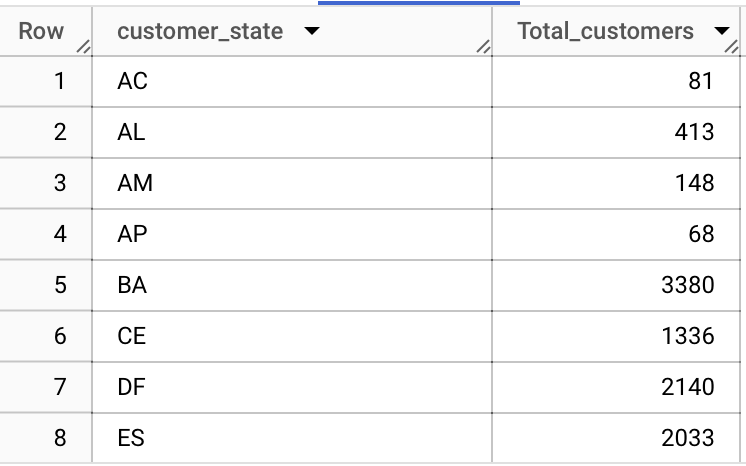
count(\*) as monthly\_orders

from `long-leaf-433203-j5.target\_sql.orders` o join `long-leaf-433203-j5.target\_sql.customers` c on o.customer\_id = c.customer\_id

group by 1,2

order by 1,3

1. How are the customers distributed across all the states?



select

customer\_state,

count(distinct customer\_id) as Total\_customers

from `target\_sql.customers`

group by 1

order by 1

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

* 1. 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.

|  |  |  |
| --- | --- | --- |
| Row | pct\_increase |  |
| 1 | 236.98 |  |

select round((b.order\_cost/a.order\_cost) \*100,2) as pct\_increase

from

(select

--format\_date('%B', o.order\_purchase\_timestamp) as month\_name,

sum(p.payment\_value) as order\_cost,

'year' as year

from `target\_sql.payments` p join `target\_sql.orders` o on

p.order\_id = o.order\_id

and extract(year from o.order\_purchase\_timestamp) in (2017)

and extract(month from o.order\_purchase\_timestamp) in (1,2,3,4,5,6,7,8)

group by 2)a

join

(select

--format\_date('%B', o.order\_purchase\_timestamp) as month\_name,

sum(p.payment\_value) as order\_cost,

'year'as year

from `target\_sql.payments` p join `target\_sql.orders` o on

p.order\_id = o.order\_id

and extract(year from o.order\_purchase\_timestamp) in (2018)

and extract(month from o.order\_purchase\_timestamp) in (1,2,3,4,5,6,7,8)

group by 2)b

on a.year = b.year

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

A screenshot of a computer

Description automatically generated

select

c.customer\_city as State,

round(sum(p.payment\_value),2) as Total\_order\_cost,

round(avg(p.payment\_value),2) as Avg\_order\_cost

from `target\_sql.customers` c join `target\_sql.orders` o on

c.customer\_id = o.customer\_id join `target\_sql.payments` p on o.order\_id = p.order\_id

group by 1

order by 1

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



select

c.customer\_city as State,

round(sum(oi.freight\_value),2) as Total\_order\_freight\_cost,

round(avg(oi.freight\_value),2) as Avg\_order\_freight\_cost

from `target\_sql.customers` c join `target\_sql.orders` o on

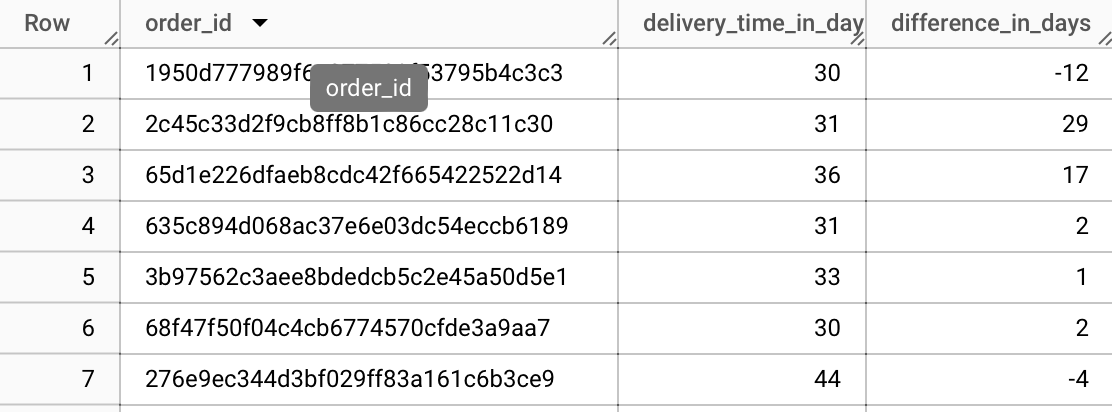
c.customer\_id = o.customer\_id join `target\_sql.order\_items` oi on oi.order\_id = o.order\_id

group by 1

order by 1

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

* 1. Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.  
     Also, calculate the difference (in days) between the estimated & actual delivery date of an order.  
     Do this in a single query.  
       
     You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
     1. **time\_to\_deliver** = order\_delivered\_customer\_date - order\_purchase\_timestamp
     2. **diff\_estimated\_delivery** = order\_delivered\_customer\_date - order\_estimated\_delivery\_date



select

order\_id,

DATE\_DIFF(

DATE(order\_delivered\_customer\_date),

DATE(order\_purchase\_timestamp),

DAY

) AS delivery\_time\_in\_days,

DATE\_DIFF(

DATE(order\_estimated\_delivery\_date),

DATE(order\_delivered\_customer\_date),

DAY

) AS difference\_in\_days

from `target\_sql.orders`

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

select \* from(

(select

c.customer\_city as State,

round(avg(oi.freight\_value),2) as Avg\_freight\_cost

from `target\_sql.customers` c join `target\_sql.orders` o on

c.customer\_id = o.customer\_id join `target\_sql.order\_items` oi on oi.order\_id = o.order\_id

group by 1

order by 2 desc

limit 5)

union all

(select

c.customer\_city as State,

round(avg(oi.freight\_value),2) as Avg\_freight\_cost

from `target\_sql.customers` c join `target\_sql.orders` o on

c.customer\_id = o.customer\_id join `target\_sql.order\_items` oi on oi.order\_id = o.order\_id

group by 1

order by 2

limit 5))t1

order by Avg\_freight\_cost

|  |  |  |
| --- | --- | --- |
| Row | State | Avg\_freight\_cost |
| 1 | curral de cima | 0 |
| 2 | caridade | 0 |
| 3 | coronel pacheco | 0 |
| 4 | aragoiania | 0 |
| 5 | capitao andrade | 0 |
| 6 | marilac | 142.49 |
| 7 | canapi | 147.32 |
| 8 | almino afonso | 170.11 |
| 9 | amarante | 193.84 |
| 10 | itupiranga | 203.38 |

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

select \* from

(

(select

c.customer\_state,

round(avg(DATE\_DIFF(

DATE(order\_delivered\_customer\_date),

DATE(order\_purchase\_timestamp),

DAY

)),2) AS Avg\_delivery\_time

from `target\_sql.customers` c join `target\_sql.orders` o on

c.customer\_id = o.customer\_id

group by 1

order by 2 desc

limit 5)

union all

(select

c.customer\_state,

round(avg(DATE\_DIFF(

DATE(order\_delivered\_customer\_date),

DATE(order\_purchase\_timestamp),

DAY

)),2) AS Avg\_delivery\_time

from `target\_sql.customers` c join `target\_sql.orders` o on

c.customer\_id = o.customer\_id

group by 1

order by 2

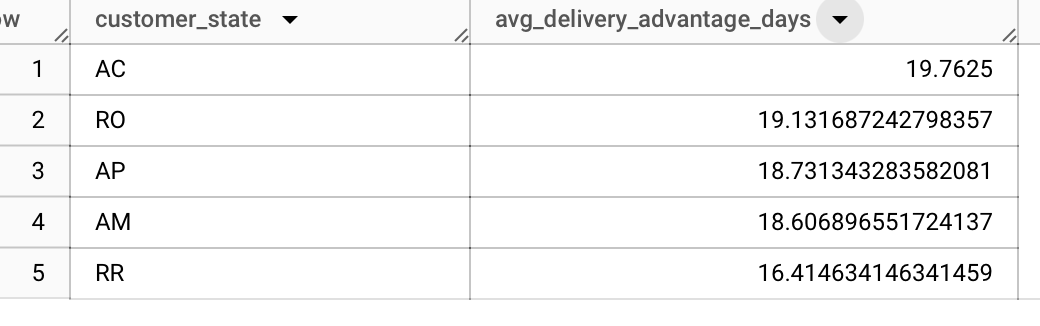
limit 5)

)a

order by Avg\_delivery\_time

|  |  |  |
| --- | --- | --- |
| Row | customer\_state | Avg\_delivery\_time |
| 1 | SP | 8.7 |
| 2 | PR | 11.94 |
| 3 | MG | 11.95 |
| 4 | DF | 12.9 |
| 5 | SC | 14.91 |
| 6 | PA | 23.73 |
| 7 | AL | 24.5 |
| 8 | AM | 26.36 |
| 9 | AP | 27.18 |
| 10 | RR | 29.34 |

* 1. 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

c.customer\_state,

AVG(DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date, DAY)) AS avg\_delivery\_advantage\_days

FROM

`target\_sql.customers` c

JOIN

`target\_sql.orders` o

ON c.customer\_id = o.customer\_id

WHERE

o.order\_delivered\_customer\_date IS NOT NULL

AND o.order\_estimated\_delivery\_date IS NOT NULL

and o.order\_status='delivered'

GROUP BY

c.customer\_state

HAVING

AVG(DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date, DAY)) > 0

ORDER BY

avg\_delivery\_advantage\_days DESC

LIMIT 5;

**6.Analysis based on the payments:**

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

A screenshot of a calendar

Description automatically generated

select

p.payment\_type,

extract(month from o.order\_purchase\_timestamp) as month,

extract(year from o.order\_purchase\_timestamp) as year,

count(o.order\_id) as month\_on\_month\_orders

from

`target\_sql.customers` c join `target\_sql.orders` o

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

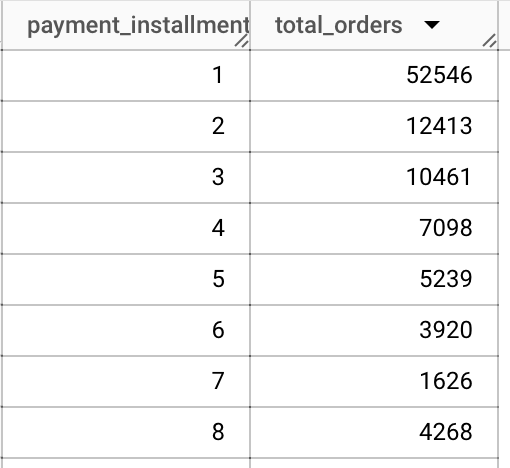
join `target\_sql.payments` p on

o.order\_id = p.order\_id

group by 1,2,3

order by 3, 2

* 1. 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 total\_orders

from

`target\_sql.orders` o join `target\_sql.payments` p on o.order\_id = p.order\_id

where p.payment\_installments > 0

group by 1

order by 2

**End of Document**