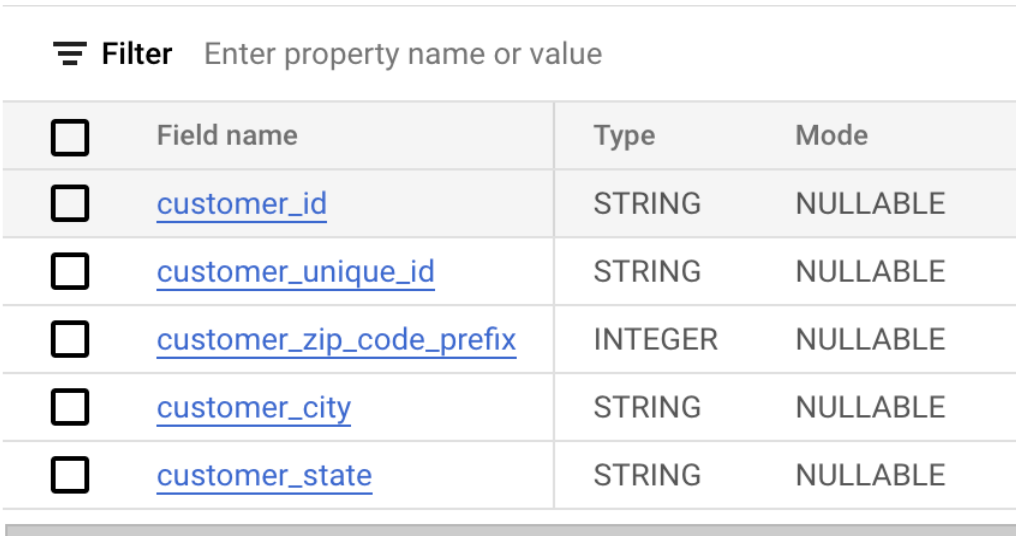
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

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

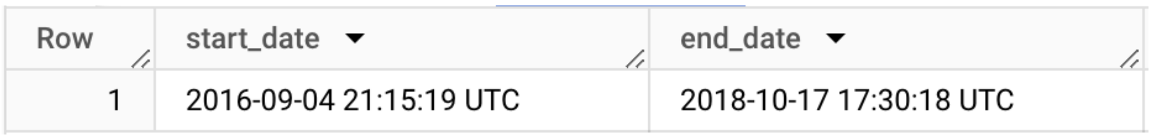


Filter Enter property name or value

| | Field name | Type | Mode |
|---|---|---|---|
| ☐ | customer_id | STRING | NULLABLE |
| ☐ | customer_unique_id | STRING | NULLABLE |
| ☐ | customer_zip_code_prefix | INTEGER | NULLABLE |
| ☐ | customer_city | STRING | NULLABLE |
| ☐ | customer_state | STRING | NULLABLE |

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

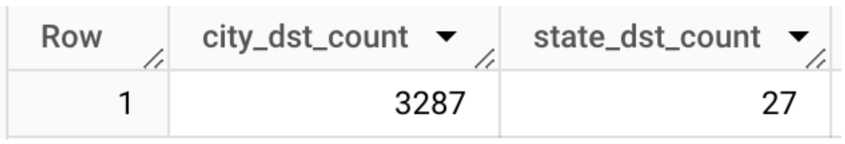
```
SELECT
MIN(order_purchase_timestamp) as start_date,
MAX(order_purchase_timestamp) as end_date
From `Target.orders`
```



| Row | start_date | end_date |
|---|---|---|
| 1 | 2016-09-04 21:15:19 UTC | 2018-10-17 17:30:18 UTC |

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

```
SELECT
count(distinct customer_city) as city_dst_count,
count(distinct customer_state) as state_dst_count
From `Target.customers` c
join `Target.orders` o
ON c.customer_id= o.customer_id
where
o.order_purchase_timestamp BETWEEN '2017-01-01' and '2017-12-31'
and
o.order_purchase_timestamp is not NULL
```

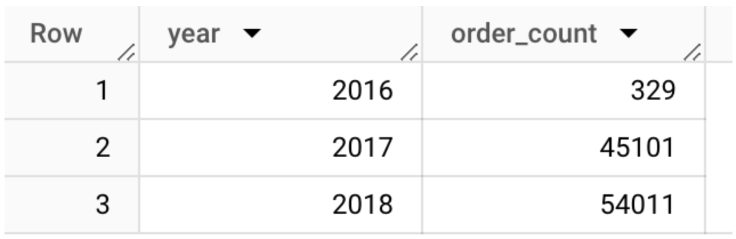


| Row | city_dst_count | state_dst_count |
|---|---|---|
| 1 | 3287 | 27 |

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,
COUNT (*) as order_count
FROM
`Target.orders`
Group By year
order by year
```

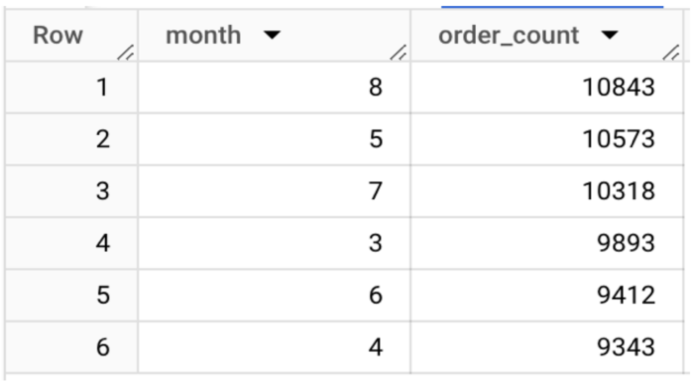


| Row | year | order_count |
|---|---|---|
| 1 | 2016 | 329 |
| 2 | 2017 | 45101 |
| 3 | 2018 | 54011 |

Insight: There is a significant increment of orders from 2016 to 2017.
Overall:  substantial growth in the number of Orders from 2016 to 2018.

**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 (*) as order_count
FROM
`Target.orders`
Group By month
order by order_count desc
limit 6
```

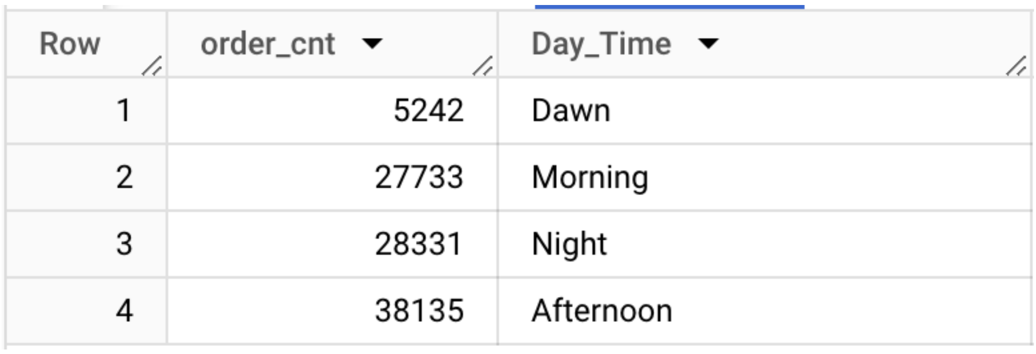


| Row | month | order_count |
|---|---|---|
| 1 | 8 | 10843 |
| 2 | 5 | 10573 |
| 3 | 7 | 10318 |
| 4 | 3 | 9893 |
| 5 | 6 | 9412 |
| 6 | 4 | 9343 |

We can see that monthly seasonality starting from March to August. Reason could be linked with some special kind of promotion or other elements which boost sales. To develop business, Marketing or any other operational planning can be done to take advantage of these months of high demand.

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

```
SELECT
count(order_id) as order_cnt,
(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 'Morning'
when EXTRACT(hour from order_purchase_timestamp) between 13 and 18 then
'Afternoon'
when EXTRACT(hour from order_purchase_timestamp) between 19 and 23 then 'Night'
END)as Day_Time
FROM `Target.orders`
Group By Day_Time
```

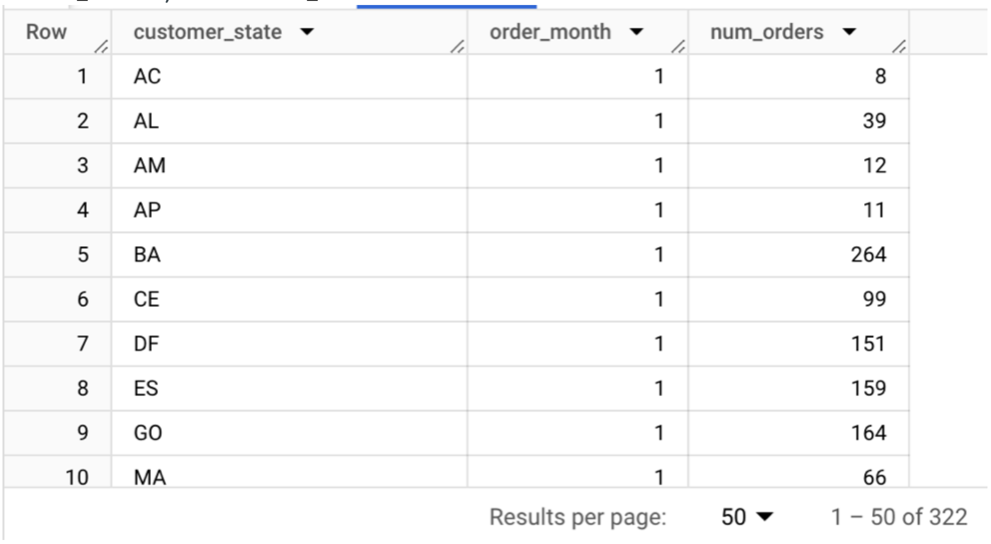


| Row | order_cnt | Day_Time |
|---|---|---|
| 1 | 5242 | Dawn |
| 2 | 27733 | Morning |
| 3 | 28331 | Night |
| 4 | 38135 | Afternoon |

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

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

```
SELECT
customer_state,
EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
COUNT(*) AS num_orders
FROM
`Target.orders` AS orders
JOIN
`Target.customers` AS customers
ON
orders.customer_id = customers.customer_id
GROUP BY
customer_state, order_month
ORDER BY
order_month, customer_state
```

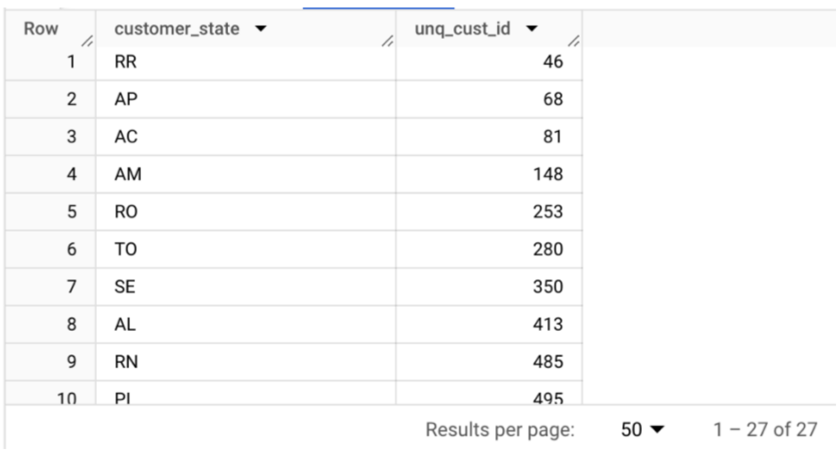


| Row | customer_state | order_month | num_orders |
|---|---|---|---|
| 1 | AC | 1 | 8 |
| 2 | AL | 1 | 39 |
| 3 | AM | 1 | 12 |
| 4 | AP | 1 | 11 |
| 5 | BA | 1 | 264 |
| 6 | CE | 1 | 99 |
| 7 | DF | 1 | 151 |
| 8 | ES | 1 | 159 |
| 9 | GO | 1 | 164 |
| 10 | MA | 1 | 66 |

Results per page: 50 1 – 50 of 322

## B. How are the customers distributed across all the states?

```
SELECT
customer_state,
count(distinct customer_id) as unq_cust_id
FROM `Target.customers`
Group By customer_state
Order By unq_cust_id asc
```



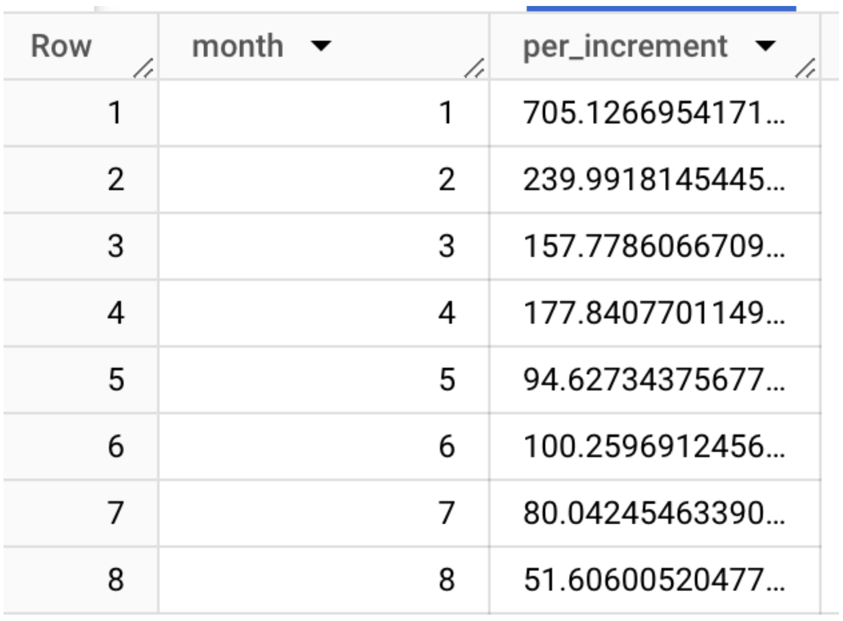
| Row | customer_state | unq_cust_id |
|---|---|---|
| 1 | RR | 46 |
| 2 | AP | 68 |
| 3 | AC | 81 |
| 4 | AM | 148 |
| 5 | RO | 253 |
| 6 | TO | 280 |
| 7 | SE | 350 |
| 8 | AL | 413 |
| 9 | RN | 485 |
| 10 | PI | 495 |

Results per page: 50 1 – 27 of 27

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).**

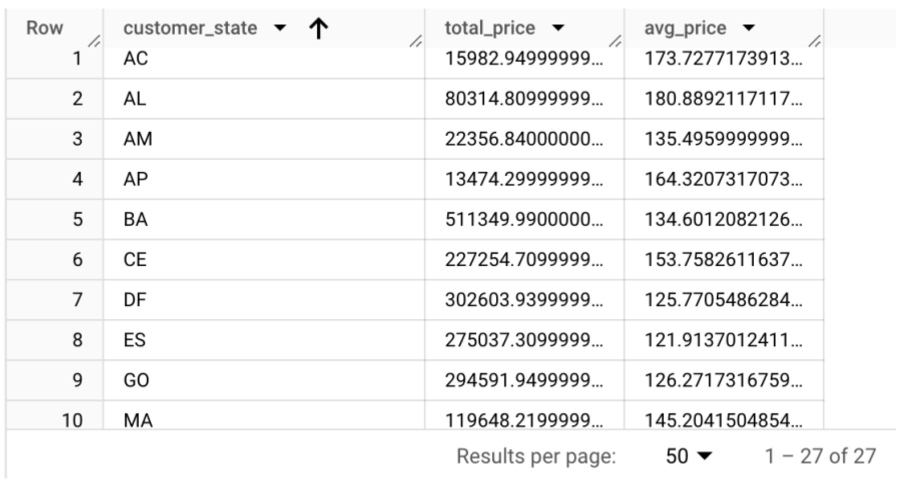
```
select
Extract (month from o.order_purchase_timestamp) as month,
((
sum (case when EXTRACT(year from o.order_purchase_timestamp)= 2018 and
extract(month from o.order_purchase_timestamp) between 1 and 8 then
p.payment_value END)
-
sum (case when EXTRACT(year from o.order_purchase_timestamp)= 2017 and
extract(month from o.order_purchase_timestamp) between 1 and 8 then
p.payment_value END
))
/
(sum (case when EXTRACT(year from o.order_purchase_timestamp)= 2017 and
extract(month  from o.order_purchase_timestamp) between 1 and 8 then
p.payment_value END))
)*100 as per_increment
from `Target.orders` o
JOIN `Target.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 month
order by month
```



| Row | month | per_increment |
|---|---|---|
| 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... |

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

```
select
c.customer_state,
sum(o_i.price) as total_price,
AVG(o_i.price) as avg_price
from `Target.orders` o
join `Target.order_items` o_i
ON o.order_id = o_i.order_id
join `Target.customers` c
ON c.customer_id = o.customer_id
Group by c.customer_state
```

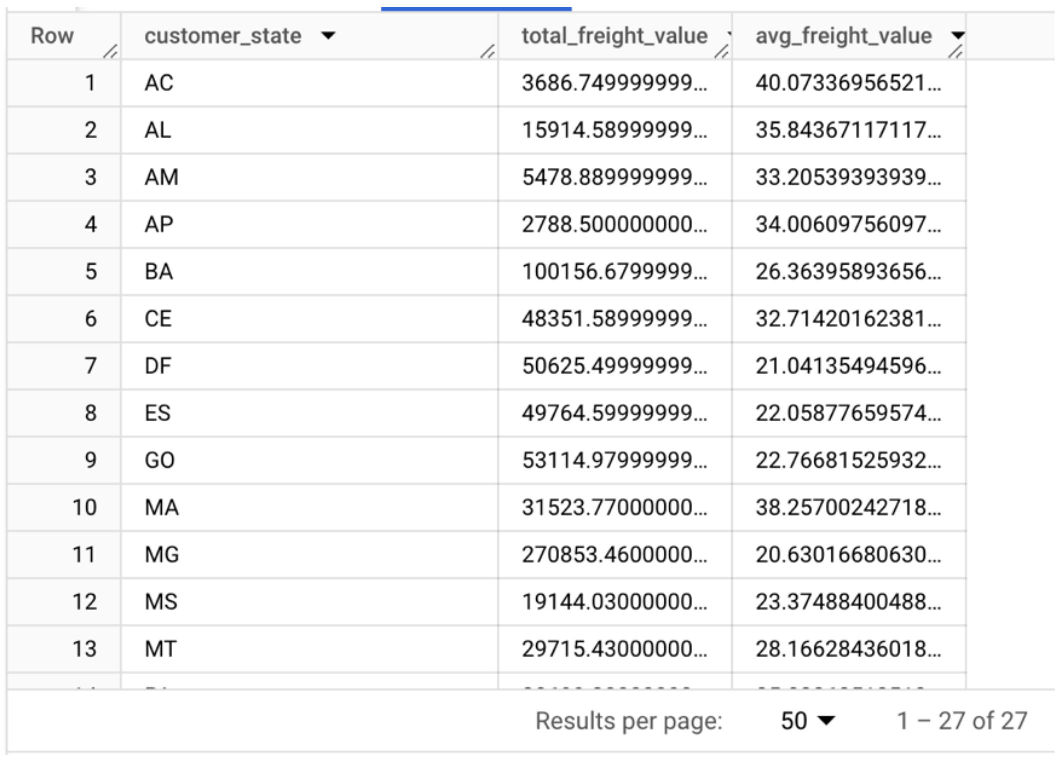


| Row | customer_state | total_price | avg_price |
|---|---|---|---|
| 1 | AC | 15982.94999999... | 173.7277173913... |
| 2 | AL | 80314.80999999... | 180.8892117117... |
| 3 | AM | 22356.84000000... | 135.4959999999... |
| 4 | AP | 13474.29999999... | 164.3207317073... |
| 5 | BA | 511349.9900000... | 134.6012082126... |
| 6 | CE | 227254.7099999... | 153.7582611637... |
| 7 | DF | 302603.9399999... | 125.7705486284... |
| 8 | ES | 275037.3099999... | 121.9137012411... |
| 9 | GO | 294591.9499999... | 126.2717316759... |
| 10 | MA | 119648.2199999... | 145.2041504854... |

Results per page: 50 ▾ 1 – 27 of 27

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

```
select
c.customer_state,
sum(o_i.freight_value) as total_freight_value,
AVG(o_i.freight_value) as avg_freight_value
from `Target.orders` o
join `Target.order_items` o_i
ON o.order_id = o_i.order_id
join `Target.customers` c
ON c.customer_id = o.customer_id
Group by c.customer_state
order by c.customer_state
```



| Row | customer_state | total_freight_value | avg_freight_value |
|---|---|---|---|
| 1 | AC | 3686.749999999... | 40.07336956521... |
| 2 | AL | 15914.58999999... | 35.84367117117... |
| 3 | AM | 5478.889999999... | 33.20539393939... |
| 4 | AP | 2788.500000000... | 34.00609756097... |
| 5 | BA | 100156.6799999... | 26.36395893656... |
| 6 | CE | 48351.58999999... | 32.71420162381... |
| 7 | DF | 50625.49999999... | 21.04135494596... |
| 8 | ES | 49764.59999999... | 22.05877659574... |
| 9 | GO | 53114.97999999... | 22.76681525932... |
| 10 | MA | 31523.77000000... | 38.25700242718... |
| 11 | MG | 270853.4600000... | 20.63016680630... |
| 12 | MS | 19144.03000000... | 23.37488400488... |
| 13 | MT | 29715.43000000... | 28.16628436018... |

Results per page: 50 1 – 27 of 27

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

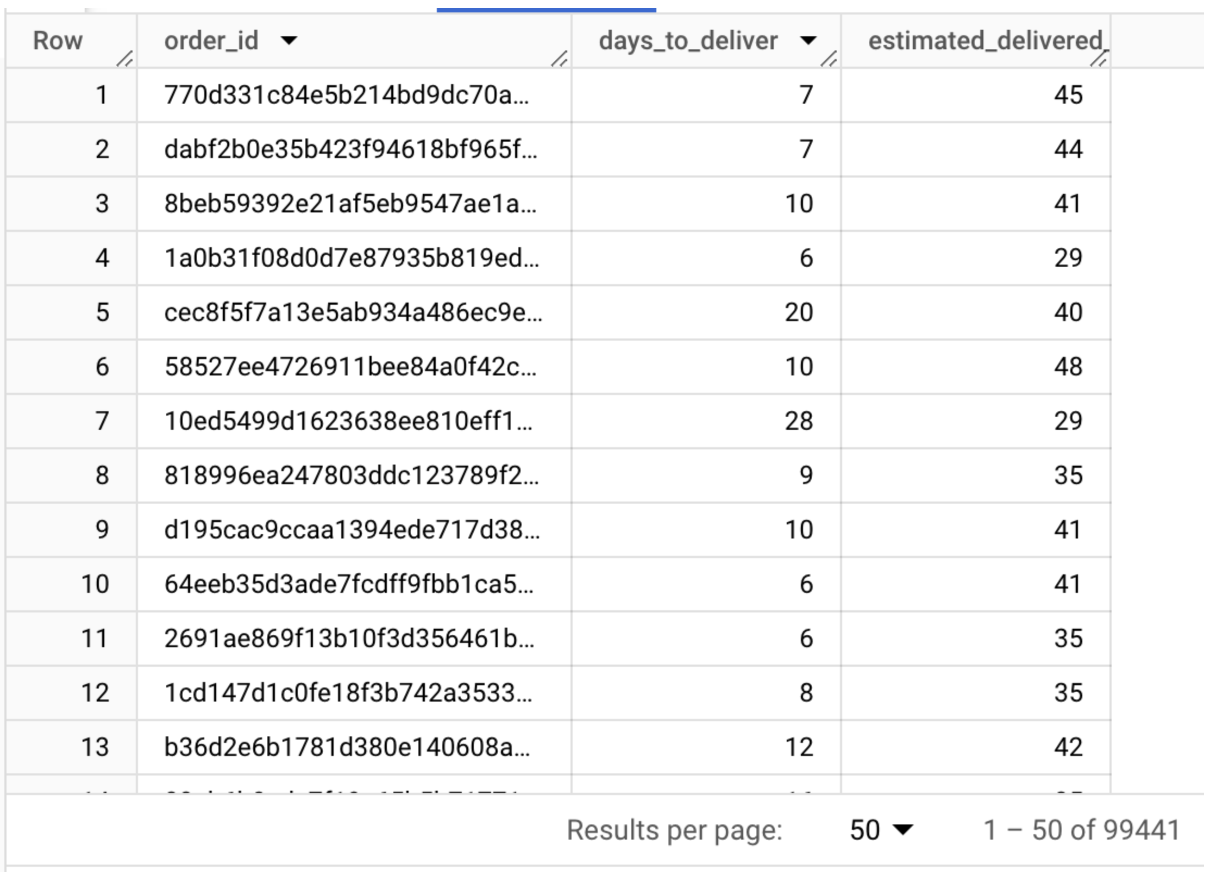
   **A. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.**
   **Also, calculate the difference (in days) between the estimated & actual delivery date of an order.**

```
SELECT
order_id,
case when order_delivered_customer_date is not NULL then
date_diff(order_delivered_customer_date,order_purchase_timestamp, day) ELSE NULL
END
as days_to_deliver,
case when order_estimated_delivery_date is not NULL and
order_delivered_customer_date is not NULL then
```

```
ABS(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day))
ELSE NULL END
as estimated_delivered_diff
FROM `Target.orders`
```

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

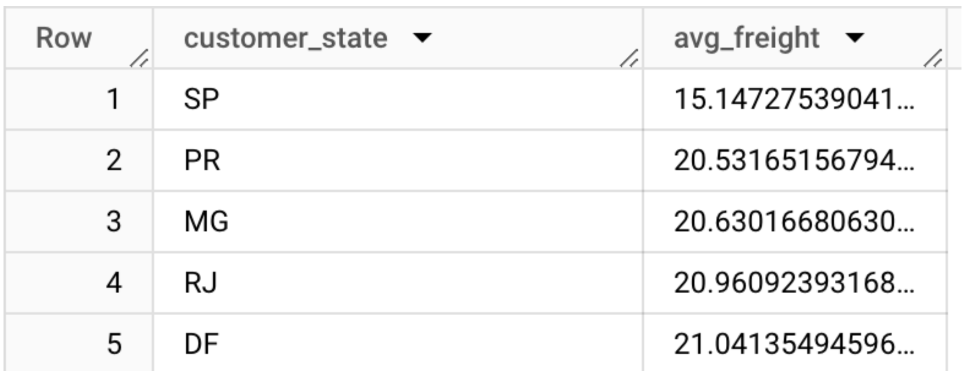
```
select
distinct c.customer_state,
avg(i.freight_value) over (partition by c.customer_state) as avg_freight
from `Target.order_items` i
join `Target.orders` o
ON i.order_id = o.order_id
join `Target.customers` c
ON o.customer_id = c.customer_id
```



| Row | order_id | days_to_deliver | estimated_delivered |
|---|---|---|---|
| 1 | 770d331c84e5b214bd9dc70a... | 7 | 45 |
| 2 | dabf2b0e35b423f94618bf965f... | 7 | 44 |
| 3 | 8beb59392e21af5eb9547ae1a... | 10 | 41 |
| 4 | 1a0b31f08d0d7e87935b819ed... | 6 | 29 |
| 5 | cec8f5f7a13e5ab934a486ec9e... | 20 | 40 |
| 6 | 58527ee4726911bee84a0f42c... | 10 | 48 |
| 7 | 10ed5499d1623638ee810eff1... | 28 | 29 |
| 8 | 818996ea247803ddc123789f2... | 9 | 35 |
| 9 | d195cac9ccaa1394ede717d38... | 10 | 41 |
| 10 | 64eeb35d3ade7fcdff9fbb1ca5... | 6 | 41 |
| 11 | 2691ae869f13b10f3d356461b... | 6 | 35 |
| 12 | 1cd147d1c0fe18f3b742a3533... | 8 | 35 |
| 13 | b36d2e6b1781d380e140608a... | 12 | 42 |

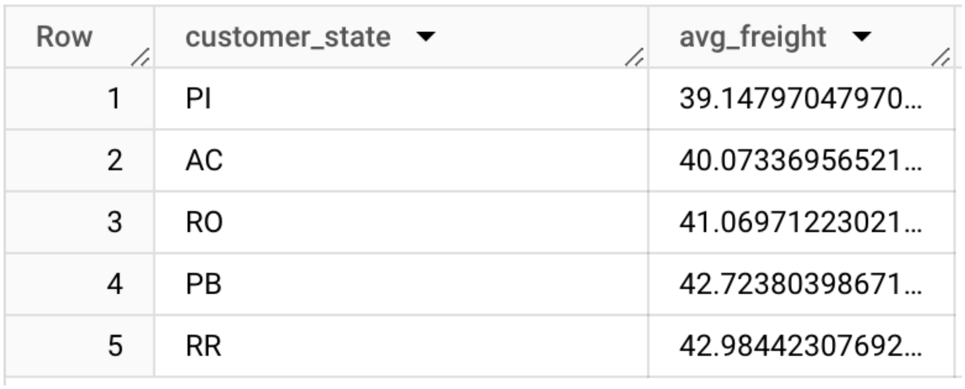
Results per page: 50 1 – 50 of 99441

```
Order By avg_freight asc
limit 5
```



| Row | customer_state | avg_freight |
|---|---|---|
| 1 | SP | 15.14727539041... |
| 2 | PR | 20.53165156794... |
| 3 | MG | 20.63016680630... |
| 4 | RJ | 20.96092393168... |
| 5 | DF | 21.04135494596... |

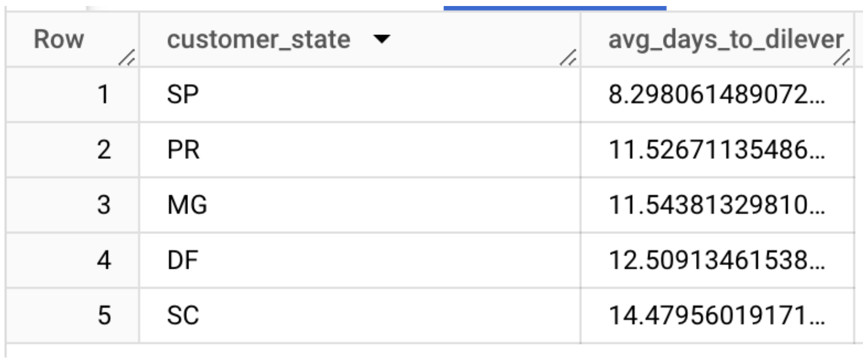
```
select
distinct c.customer_state,
avg(i.freight_value) over (partition by c.customer_state) as avg_freight
from `Target.order_items` i
join `Target.orders` o
ON i.order_id = o.order_id
join `Target.customers` c
ON o.customer_id = c.customer_id
Order By avg_freight asc
limit 5
offset 22
```



| Row | customer_state | avg_freight |
|---|---|---|
| 1 | PI | 39.14797047970... |
| 2 | AC | 40.07336956521... |
| 3 | RO | 41.06971223021... |
| 4 | PB | 42.72380398671... |
| 5 | RR | 42.98442307692... |

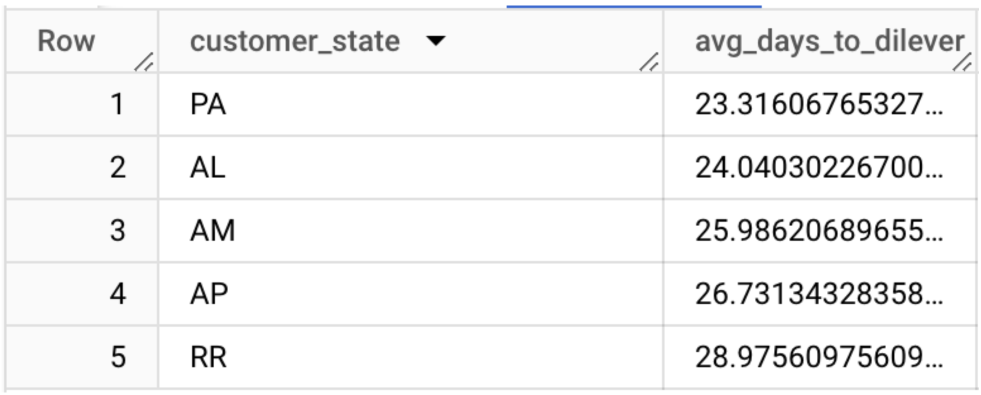
**C. Find out the top 5 states with the highest & lowest average delivery time.**

```
select
c.customer_state,
avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,
day)) as avg_days_to_dilever
FROM `Target.orders` o
JOIN `Target.customers` c
ON o.customer_id = c.customer_id
where o.order_delivered_customer_date is not NULL
Group BY c.customer_state
order by avg_days_to_dilever asc
limit 5
```



| Row | customer_state | avg_days_to_dilever |
|---|---|---|
| 1 | SP | 8.298061489072... |
| 2 | PR | 11.52671135486... |
| 3 | MG | 11.54381329810... |
| 4 | DF | 12.50913461538... |
| 5 | SC | 14.47956019171... |

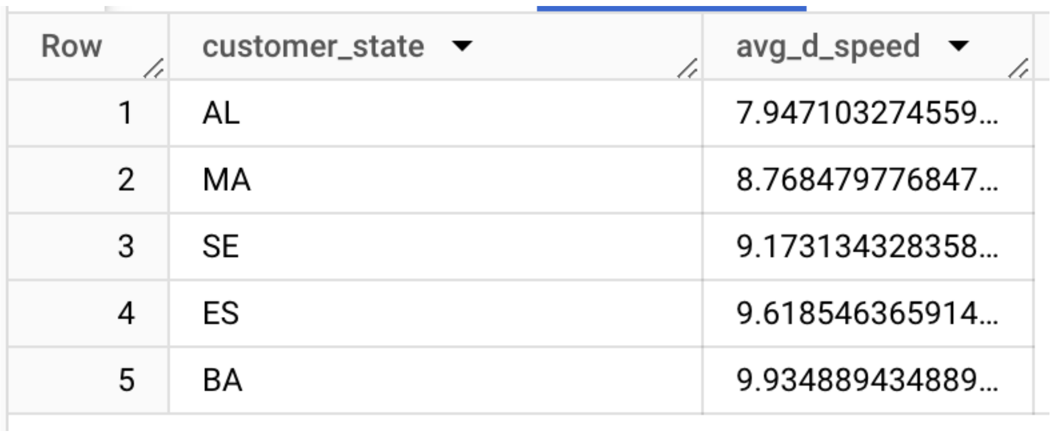
```
select
c.customer_state,
avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,
day)) as avg_days_to_dilever
FROM `Target.orders` o
JOIN `Target.customers` c
ON o.customer_id = c.customer_id
where o.order_delivered_customer_date is not NULL
Group BY c.customer_state
order by avg_days_to_dilever asc
limit 5
offset 22
```



| Row | customer_state | avg_days_to_dilever |
|---|---|---|
| 1 | PA | 23.31606765327... |
| 2 | AL | 24.04030226700... |
| 3 | AM | 25.98620689655... |
| 4 | AP | 26.73134328358... |
| 5 | RR | 28.97560975609... |

**D. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**

```
SELECT
c.customer_state,
AVG(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date,
day)) as avg_d_speed
FROM
`Target.orders` o
JOIN `Target.customers` c
ON o.customer_id = c.customer_id
Where o.order_delivered_customer_date IS NOT NULL
Group By c.customer_state
Order By avg_d_speed ASC
limit 5
```

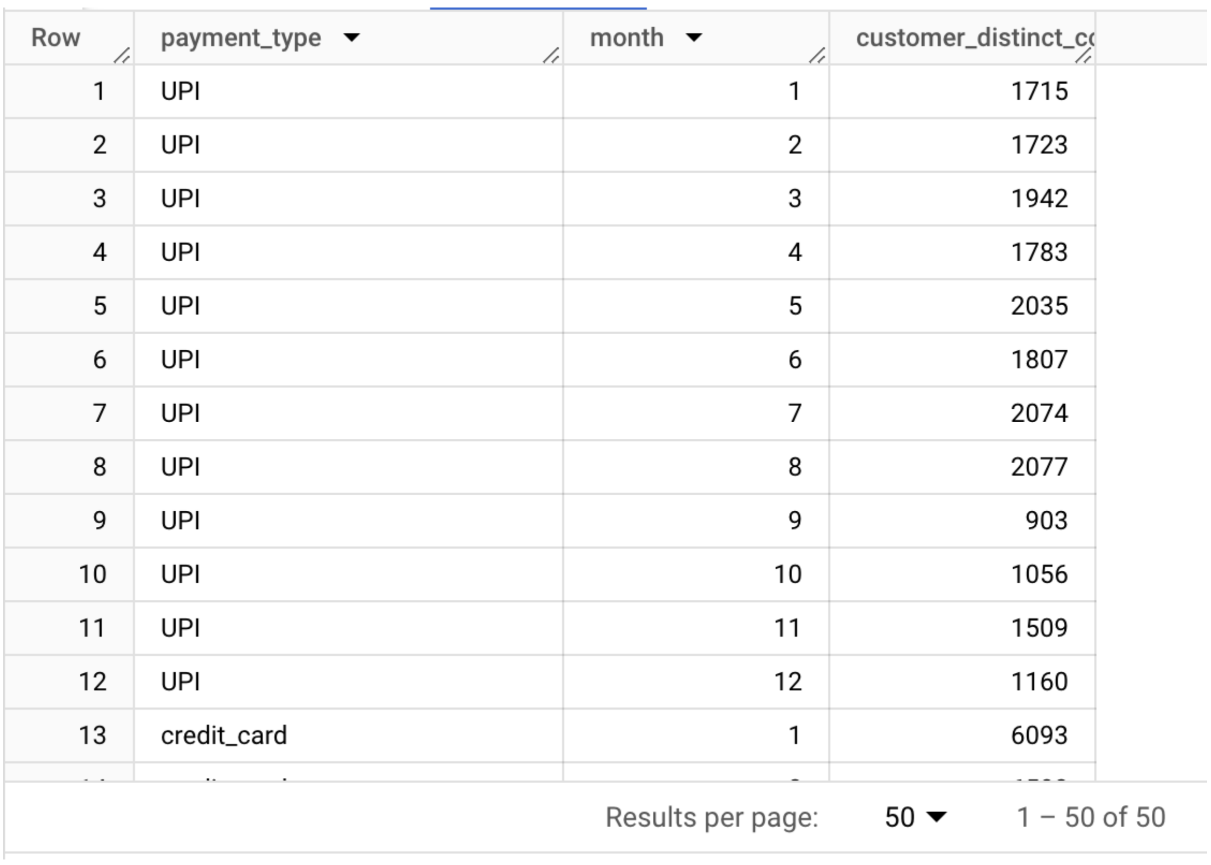


| Row | customer_state | avg_d_speed |
|---|---|---|
| 1 | AL | 7.947103274559... |
| 2 | MA | 8.768479776847... |
| 3 | SE | 9.173134328358... |
| 4 | ES | 9.618546365914... |
| 5 | BA | 9.934889434889... |

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 o.order_purchase_timestamp) as month,
count(distinct o.order_id) as customer_distinct_count
from `Target.payments` p
join `Target.orders` o
on p.order_id = o.order_id
Group By p.payment_type, month
Order By p.payment_type, month
```

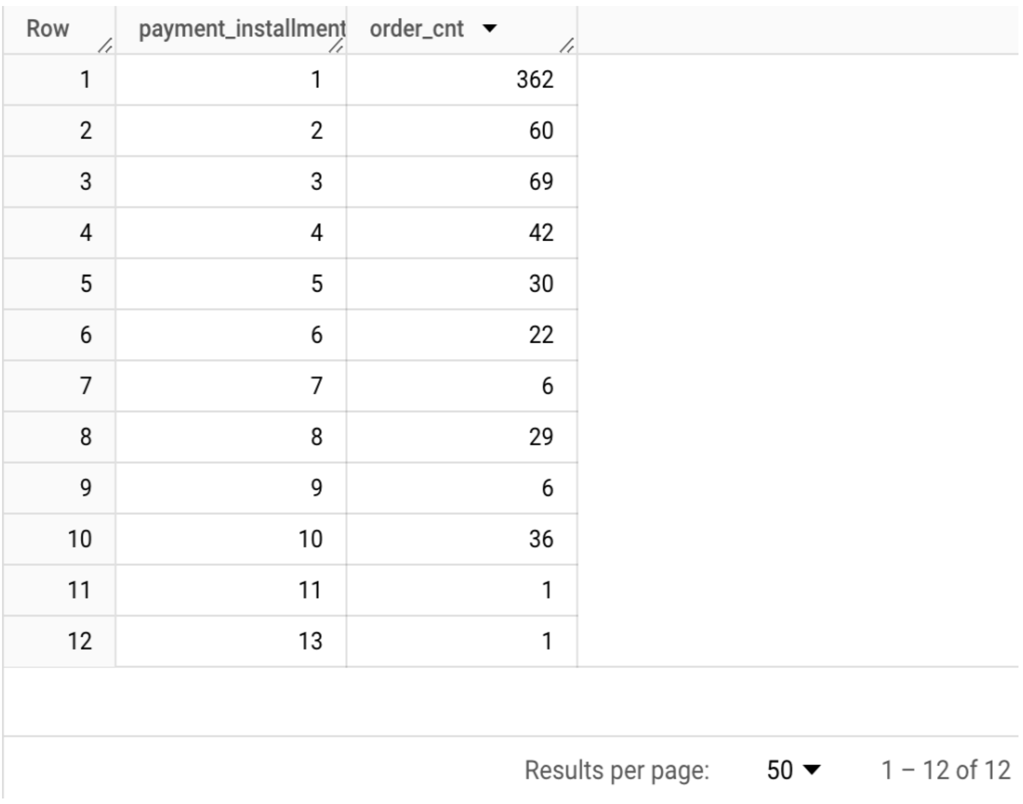


| Row | payment_type | month | customer_distinct_c |
|---|---|---|---|
| 1 | UPI | 1 | 1715 |
| 2 | UPI | 2 | 1723 |
| 3 | UPI | 3 | 1942 |
| 4 | UPI | 4 | 1783 |
| 5 | UPI | 5 | 2035 |
| 6 | UPI | 6 | 1807 |
| 7 | UPI | 7 | 2074 |
| 8 | UPI | 8 | 2077 |
| 9 | UPI | 9 | 903 |
| 10 | UPI | 10 | 1056 |
| 11 | UPI | 11 | 1509 |
| 12 | UPI | 12 | 1160 |
| 13 | credit_card | 1 | 6093 |

Results per page: 50 1 – 50 of 50

**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_cnt
from `Target.orders` o
Join `Target.payments` p
ON o.order_id = p.order_id
where order_status = 'canceled'
Group By p.payment_installments
order by p.payment_installments
```



| Row | payment_installment | order_cnt ▼ |
|---|---|---|
| 1 | 1 | 362 |
| 2 | 2 | 60 |
| 3 | 3 | 69 |
| 4 | 4 | 42 |
| 5 | 5 | 30 |
| 6 | 6 | 22 |
| 7 | 7 | 6 |
| 8 | 8 | 29 |
| 9 | 9 | 6 |
| 10 | 10 | 36 |
| 11 | 11 | 1 |
| 12 | 13 | 1 |

Results per page: 50 ▼ 1 – 12 of 12