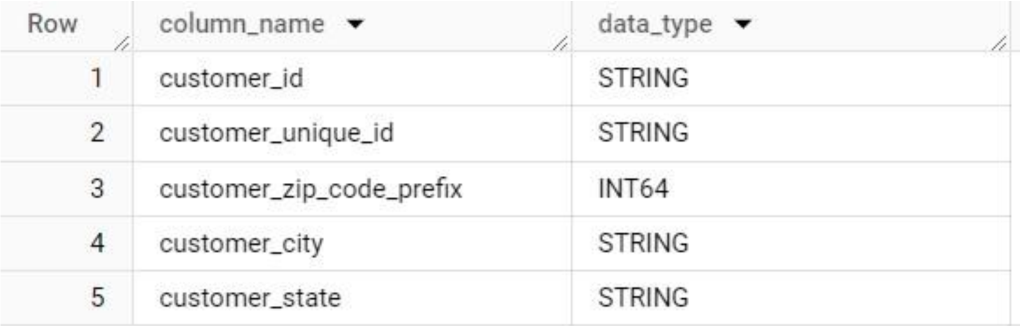
# Unpacking Retailer's Journey in Brazil: A Deep Dive into 100,000 Orders and Customer Behaviour (2016-2018) using SQL

## I. 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.

Query –>

```
 SELECT
column_name,
data_type
 FROM
 `scaler-dsml-sql-433115.target.INFORMATION_SCHEMA.COLUMNS`
WHERE  table_name = 'customers';
```

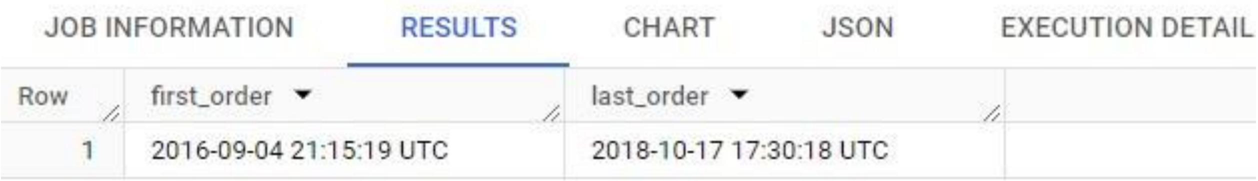


| Row | column_name | data_type |
|---|---|---|
| 1 | customer_id | STRING |
| 2 | customer_unique_id | STRING |
| 3 | customer_zip_code_prefix | INT64 |
| 4 | customer_city | STRING |
| 5 | customer_state | STRING |

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

Query->

```
SELECT
MIN(order_purchase_timestamp) AS first_order,
MAX(order_purchase_timestamp) AS
last_order FROM target.orders
```

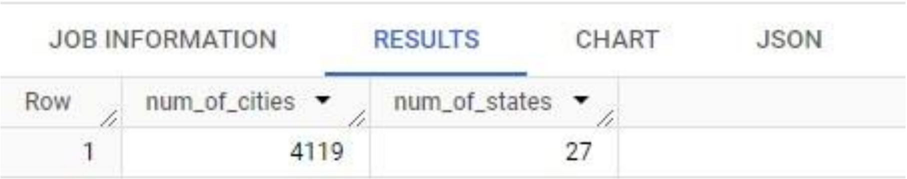


| Row | first_order | last_order |
|---|---|---|
| 1 | 2016-09-04 21:15:19 UTC | 2018-10-17 17:30:18 UTC |

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

Query->

```
SELECT
  COUNT(DISTINCT customer_city) AS num_of_cities,
  COUNT(DISTINCT customer_state) AS
num_of_states  FROM   target.orders o
JOIN
  `target.customers` c
ON
  o.customer_id = c.customer_id
```



| Row | num_of_cities | num_of_states |
|---|---|---|
| 1 | 4119 | 27 |

Insights -

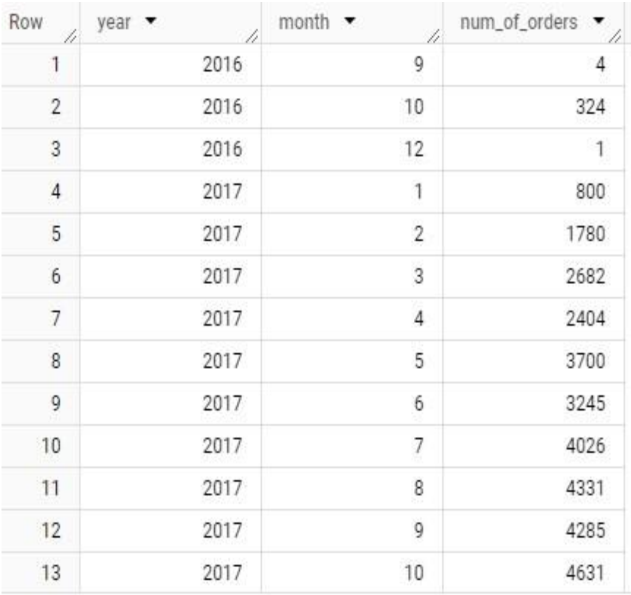
- *First order was placed on 4th sep 2016 and last order was placed On 17th october 2018. Which mean target was operational in Brazil for 2 years 1 month 11 days approx.*
- *From the above queries we can say that target has its customers from All the 27 states of Brazil.*

## II. In-depth Exploration:

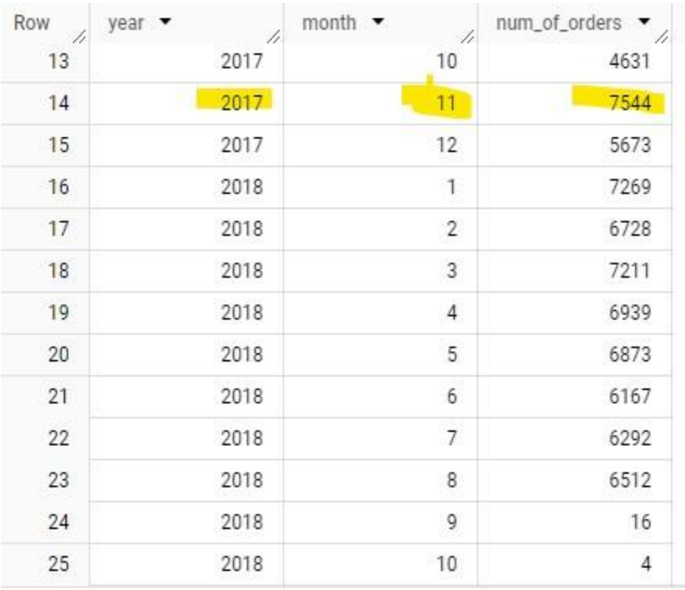
### A. Is there a growing trend in the no. of orders placed over the past years?

Query –>

```
SELECT
  EXTRACT(year   FROM
order_purchase_timestamp) AS year,
  EXTRACT(month   FROM
order_purchase_timestamp) AS month,
  COUNT(order_id) AS
num_of_orders FROM
target.orders GROUP BY 1,2 ORDER
BY 1,2
```



| Row | year | month | num_of_orders |
|---|---|---|---|
| 1 | 2016 | 9 | 4 |
| 2 | 2016 | 10 | 324 |
| 3 | 2016 | 12 | 1 |
| 4 | 2017 | 1 | 800 |
| 5 | 2017 | 2 | 1780 |
| 6 | 2017 | 3 | 2682 |
| 7 | 2017 | 4 | 2404 |
| 8 | 2017 | 5 | 3700 |
| 9 | 2017 | 6 | 3245 |
| 10 | 2017 | 7 | 4026 |
| 11 | 2017 | 8 | 4331 |
| 12 | 2017 | 9 | 4285 |
| 13 | 2017 | 10 | 4631 |



| Row | year | month | num_of_orders |
|---|---|---|---|
| 13 | 2017 | 10 | 4631 |
| 14 | 2017 | 11 | 7544 |
| 15 | 2017 | 12 | 5673 |
| 16 | 2018 | 1 | 7269 |
| 17 | 2018 | 2 | 6728 |
| 18 | 2018 | 3 | 7211 |
| 19 | 2018 | 4 | 6939 |
| 20 | 2018 | 5 | 6873 |
| 21 | 2018 | 6 | 6167 |
| 22 | 2018 | 7 | 6292 |
| 23 | 2018 | 8 | 6512 |
| 24 | 2018 | 9 | 16 |
| 25 | 2018 | 10 | 4 |

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

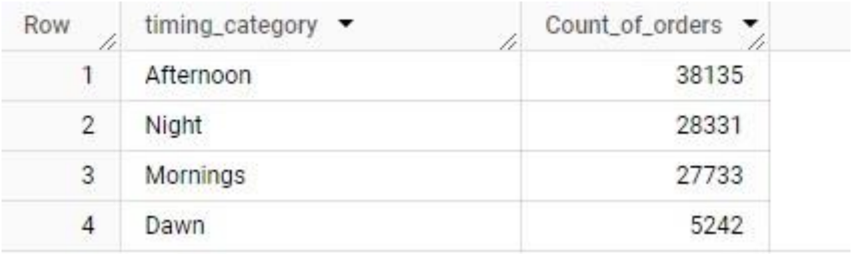
❾ *We see maximum sales in year end (last three months and beginning month) due to various festivals.*

❾ *we can see maximum sales in November of 2017 because of Black Friday(Black Friday is a major shopping day that's celebrated for a number of reasons).*

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

Query->

```
SELECT timing_category,count(order_id) Count_of_orders FROM(
     SELECT   order_id,
       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'         when extract(hour from
     order_purchase_timestamp) between 19 and 23 then
     'Night'       end as        timing_category FROM
     target.orders
     ) group by timing_category
     Order by Count_of_orders desc
```



| Row | timing_category | Count_of_orders |
|---|---|---|
| 1 | Afternoon | 38135 |
| 2 | Night | 28331 |
| 3 | Mornings | 27733 |
| 4 | Dawn | 5242 |

Insights -

- *Peak sale was found during year end (last three month and first month) in 2017-18 as per data it also seems target was fully operational during year 2017-18 hence we can see Month on Month growth in number of orders placed.*
- *As per the query results above we can see Peak sale during Afternoon •*

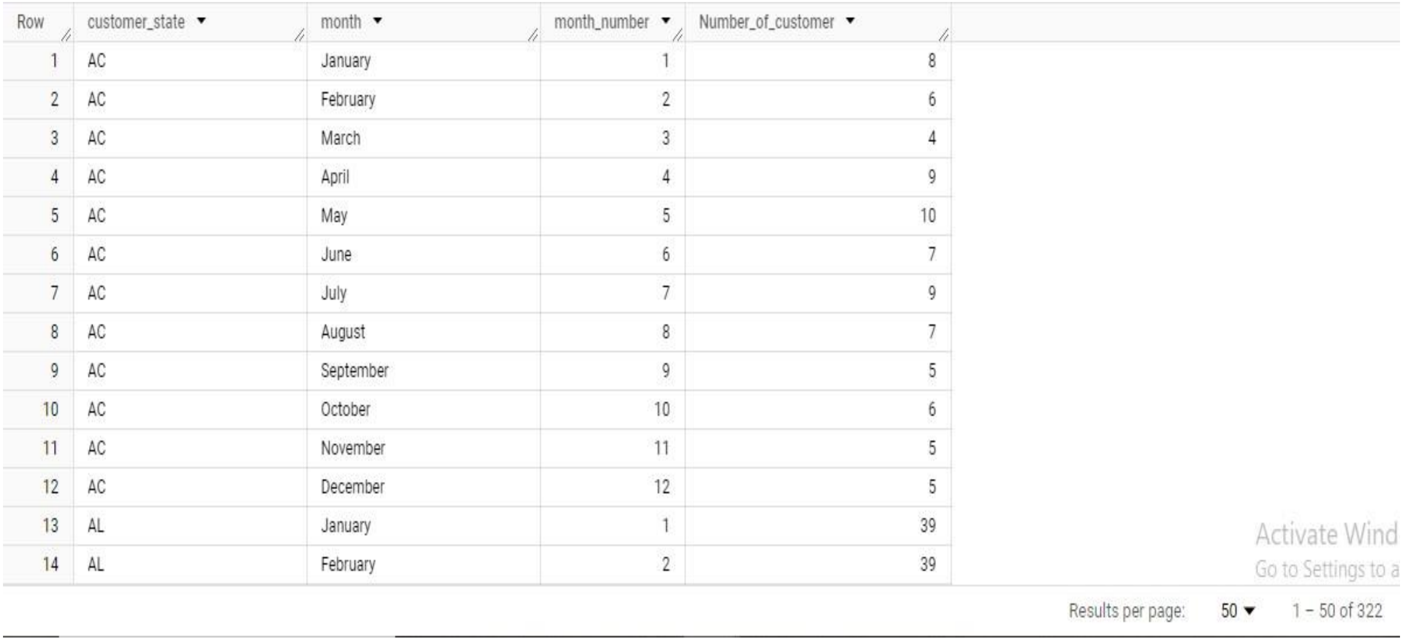
*Resources Should be aligned in Afternoon accordingly*

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

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

Query->

```
SELECT   customer_state,
format_datetime('%B',order_purchase_timestamp) as month,
extract(month from order_purchase_timestamp) as month_number
  ,count(c.customer_id) as Number_of_customer
FROM
  `target.orders` o
JOIN
  `target.customers` c
ON
  o.customer_id = c.customer_id group
by 1,2,3 order by 1,3
```



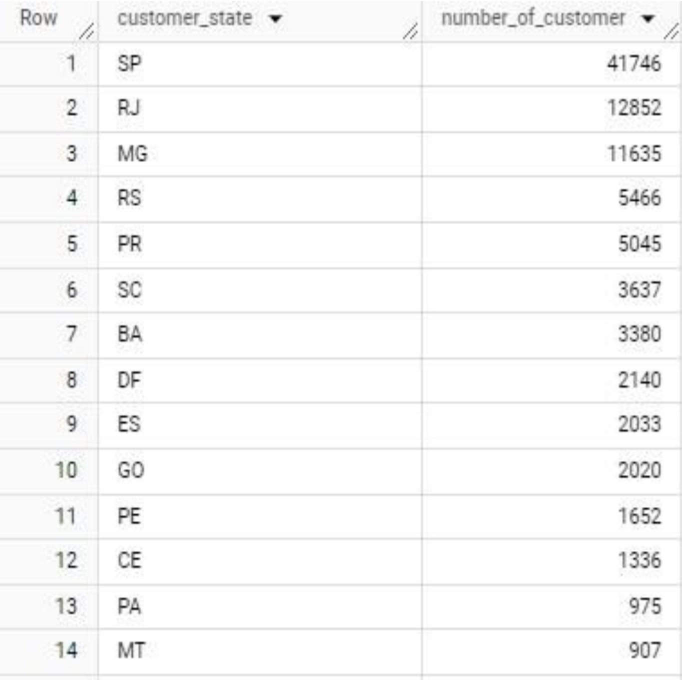
| Row | customer_state | month | month_number | Number_of_customer |
|---|---|---|---|---|
| 1 | AC | January | 1 | 8 |
| 2 | AC | February | 2 | 6 |
| 3 | AC | March | 3 | 4 |
| 4 | AC | April | 4 | 9 |
| 5 | AC | May | 5 | 10 |
| 6 | AC | June | 6 | 7 |
| 7 | AC | July | 7 | 9 |
| 8 | AC | August | 8 | 7 |
| 9 | AC | September | 9 | 5 |
| 10 | AC | October | 10 | 6 |
| 11 | AC | November | 11 | 5 |
| 12 | AC | December | 12 | 5 |
| 13 | AL | January | 1 | 39 |
| 14 | AL | February | 2 | 39 |

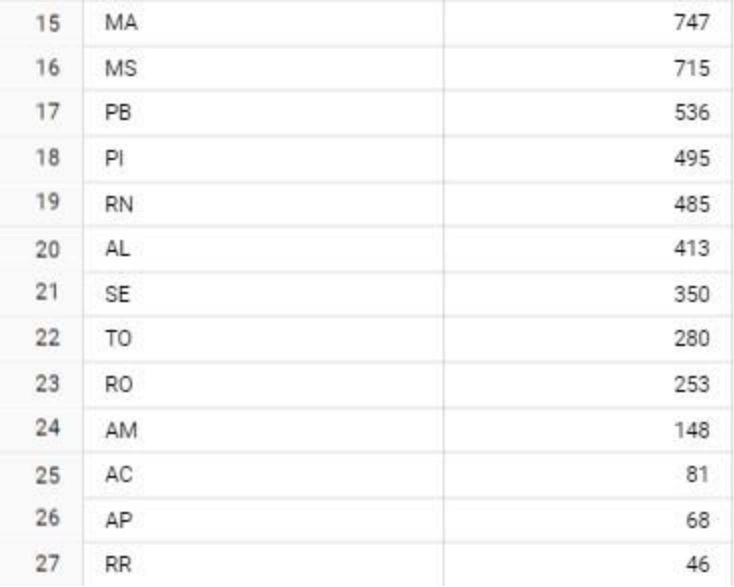
Results per page: 50 ▾ 1 – 50 of 322

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

Query->

```
SELECT
customer_state,
  COUNT(DISTINCT c.customer_id) number_of_customer
FROM
  `target.customers` c
GROUP BY
customer_state
ORDER BY
  2 desc
```





| Row | customer_state | number_of_customer |
|---|---|---|
| 1 | SP | 41746 |
| 2 | RJ | 12852 |
| 3 | MG | 11635 |
| 4 | RS | 5466 |
| 5 | PR | 5045 |
| 6 | SC | 3637 |
| 7 | BA | 3380 |
| 8 | DF | 2140 |
| 9 | ES | 2033 |
| 10 | GO | 2020 |
| 11 | PE | 1652 |
| 12 | CE | 1336 |
| 13 | PA | 975 |
| 14 | MT | 907 |
| 15 | MA | 747 |
| 16 | MS | 715 |
| 17 | PB | 536 |
| 18 | PI | 495 |
| 19 | RN | 485 |
| 20 | AL | 413 |
| 21 | SE | 350 |
| 22 | TO | 280 |
| 23 | RO | 253 |
| 24 | AM | 148 |
| 25 | AC | 81 |
| 26 | AP | 68 |
| 27 | RR | 46 |

Insights -

- *Maximum number of Customers are from  state SP (Sau Paulo) i.e 41746*
- *Minimum number of Customers are from state RR (Roraima)  i.e 46*
- *There are zero orders for Roraima in August and December Reason(Roraima's population makes up about **0.3%** of Brazil's total population).*
- *lowest customer activity is Roraima (RR)**,** with 46 customers, followed by Amapa (AP) and Acre (AC) Localized Marketing Campaigns, Partner with Local Businesses can help is increasing customer base in lowest customer activity areas*

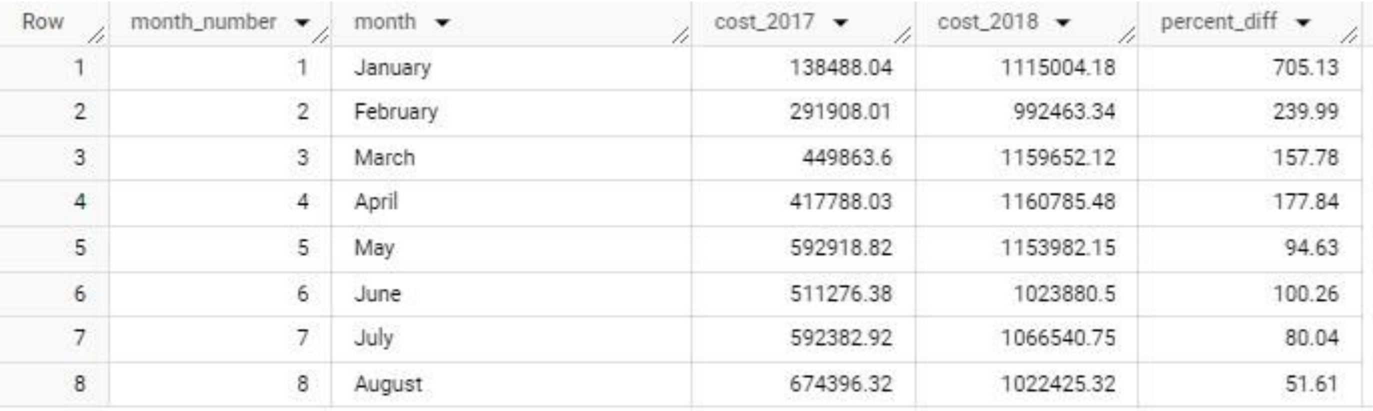
## IV. Impact on Economy: Analyze the money movement by ecommerce by looking at order prices, freight and others.

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

Query –>

```
SELECT
  *,
  ROUND(((cost_2018-cost_2017)/cost_2017)*100,2) AS percent_diff
FROM (
  SELECT
    EXTRACT(MONTH     FROM
order_purchase_timestamp) AS month_number,
    FORMAT_DATETIME('%B', order_purchase_timestamp) AS month,
    ROUND(SUM(CASE
          WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017
THEN p.payment_value
          ELSE 0
      END
        ),2) AS cost_2017,
    ROUND(SUM(CASE
          WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2018
THEN p.payment_value
          ELSE 0
      END
        ),2) AS cost_2018
FROM `target.orders` o join `target.payments` p
```

```
    ON
      o.order_id = p.order_id
    WHERE
      EXTRACT(MONTH     FROM
  order_purchase_timestamp) < 9
      AND EXTRACT(YEAR     FROM
  order_purchase_timestamp) BETWEEN 2017
      AND 2018
  GROUP BY
  month_number,
  month   ORDER BY
  month_number )
```



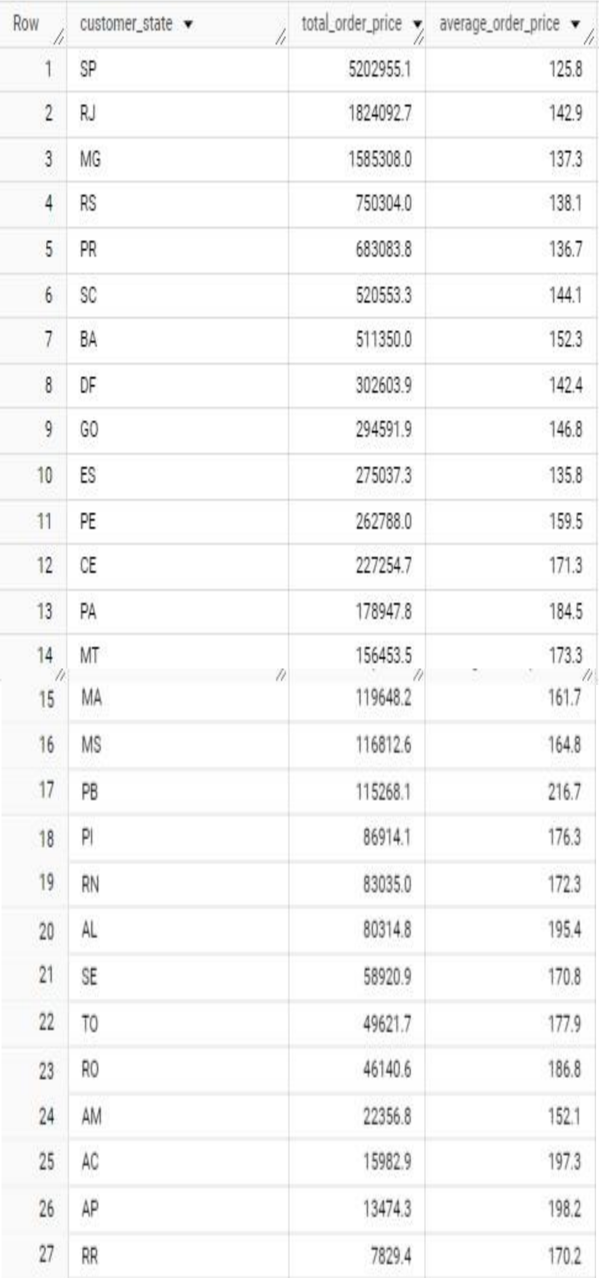
| Row | month_number | month | cost_2017 | cost_2018 | percent_diff |
|---|---|---|---|---|---|
| 1 | 1 | January | 138488.04 | 1115004.18 | 705.13 |
| 2 | 2 | February | 291908.01 | 992463.34 | 239.99 |
| 3 | 3 | March | 449863.6 | 1159652.12 | 157.78 |
| 4 | 4 | April | 417788.03 | 1160785.48 | 177.84 |
| 5 | 5 | May | 592918.82 | 1153982.15 | 94.63 |
| 6 | 6 | June | 511276.38 | 1023880.5 | 100.26 |
| 7 | 7 | July | 592382.92 | 1066540.75 | 80.04 |
| 8 | 8 | August | 674396.32 | 1022425.32 | 51.61 |

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

Query->

```
  SELECT c.customer_state,
    ROUND(SUM(order_total_price),1) AS total_order_price,
  ROUND(AVG(order_total_price),1) AS average_order_price
```

```
FROM (     SELECT
oi.order_id,
    SUM(oi.price) AS
order_total_price FROM
target.order_items oi GROUP BY
oi.order_id
) order_totals JOIN
target.orders o
 ON order_totals.order_id =
o.order_id JOIN
target.customers c
 ON o.customer_id =
c.customer_id
GROUP BY
   c.customer_state order by 2
desc,3 desc
```



| Row | customer_state | total_order_price | average_order_price |
|---|---|---|---|
| 1 | SP | 5202955.1 | 125.8 |
| 2 | RJ | 1824092.7 | 142.9 |
| 3 | MG | 1585308.0 | 137.3 |
| 4 | RS | 750304.0 | 138.1 |
| 5 | PR | 683083.8 | 136.7 |
| 6 | SC | 520553.3 | 144.1 |
| 7 | BA | 511350.0 | 152.3 |
| 8 | DF | 302603.9 | 142.4 |
| 9 | GO | 294591.9 | 146.8 |
| 10 | ES | 275037.3 | 135.8 |
| 11 | PE | 262788.0 | 159.5 |
| 12 | CE | 227254.7 | 171.3 |
| 13 | PA | 178947.8 | 184.5 |
| 14 | MT | 156453.5 | 173.3 |
| 15 | MA | 119648.2 | 161.7 |
| 16 | MS | 116812.6 | 164.8 |
| 17 | PB | 115268.1 | 216.7 |
| 18 | PI | 86914.1 | 176.3 |
| 19 | RN | 83035.0 | 172.3 |
| 20 | AL | 80314.8 | 195.4 |
| 21 | SE | 58920.9 | 170.8 |
| 22 | TO | 49621.7 | 177.9 |
| 23 | RO | 46140.6 | 186.8 |
| 24 | AM | 22356.8 | 152.1 |
| 25 | AC | 15982.9 | 197.3 |
| 26 | AP | 13474.3 | 198.2 |
| 27 | RR | 7829.4 | 170.2 |

C. Calculate the Total & Average value of order freight for each state. Query->

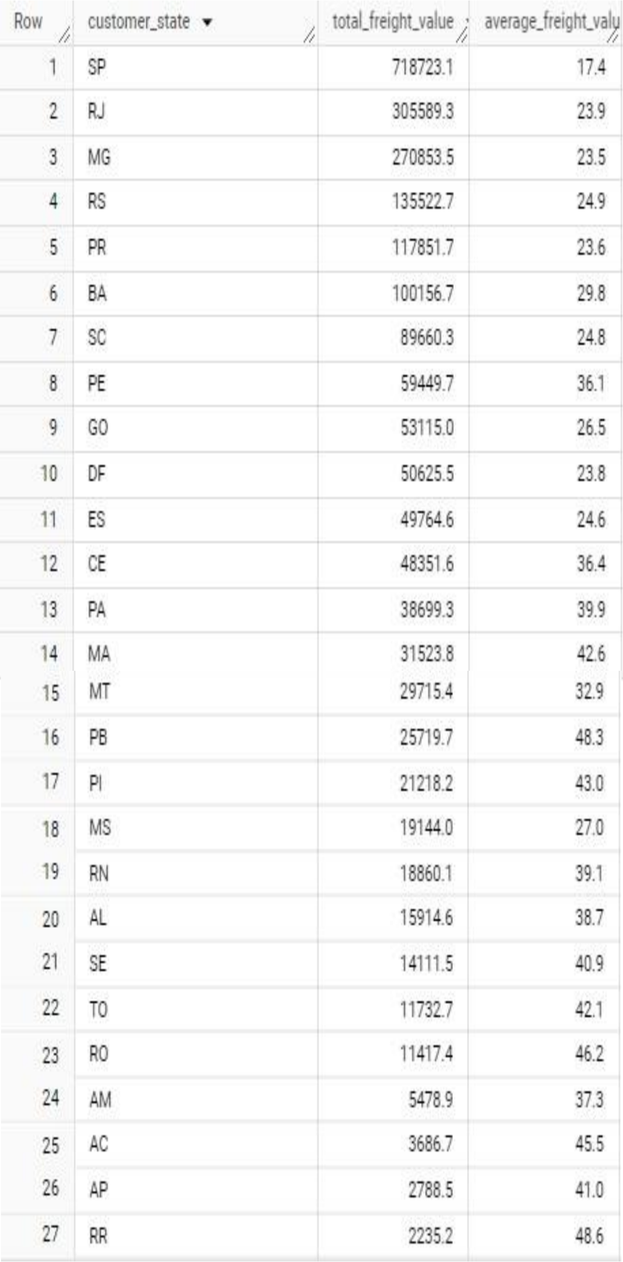
```
SELECT
    c.customer_state,
    ROUND(SUM(order_total_freight),1) AS total_freight_value,
ROUND(AVG(order_total_freight),1) AS average_freight_value
```

```
FROM (   SELECT
oi.order_id,

SUM(oi.freight_value)
     AS
order_total_freight
target.order_items oi
oi.order_id
) order_freight_totals
target.orders o
    ON
```

```
FROM
GROUP BY

JOIN
```



| Row | customer_state | total_freight_value | average_freight_valu |
|---|---|---|---|
| 1 | SP | 718723.1 | 17.4 |
| 2 | RJ | 305589.3 | 23.9 |
| 3 | MG | 270853.5 | 23.5 |
| 4 | RS | 135522.7 | 24.9 |
| 5 | PR | 117851.7 | 23.6 |
| 6 | BA | 100156.7 | 29.8 |
| 7 | SC | 89660.3 | 24.8 |
| 8 | PE | 59449.7 | 36.1 |
| 9 | GO | 53115.0 | 26.5 |
| 10 | DF | 50625.5 | 23.8 |
| 11 | ES | 49764.6 | 24.6 |
| 12 | CE | 48351.6 | 36.4 |
| 13 | PA | 38699.3 | 39.9 |
| 14 | MA | 31523.8 | 42.6 |
| 15 | MT | 29715.4 | 32.9 |
| 16 | PB | 25719.7 | 48.3 |
| 17 | PI | 21218.2 | 43.0 |
| 18 | MS | 19144.0 | 27.0 |
| 19 | RN | 18860.1 | 39.1 |
| 20 | AL | 15914.6 | 38.7 |
| 21 | SE | 14111.5 | 40.9 |
| 22 | TO | 11732.7 | 42.1 |
| 23 | RO | 11417.4 | 46.2 |
| 24 | AM | 5478.9 | 37.3 |
| 25 | AC | 3686.7 | 45.5 |
| 26 | AP | 2788.5 | 41.0 |
| 27 | RR | 2235.2 | 48.6 |

```
order_freight_totals.order_id
     = o.order_id JOIN      target.customers c
    ON o.customer_id = c.customer_id
GROUP BY
    c.customer_state order by 2 desc,3 desc
```

## Insights -

- *Maximum % increase in the cost of orders from year 2017 to 2018 can be seen in month of January by (705.13%) as in 2017 target was setting its operations and in 2018 it was fully functional.*
- *Less the total order price more the average order price and vice-versa can be seen through second query.*
- *More the value of total freight lesser is the value of average freight and viceversa as Transportation cost decreases when quantity or value of orders increases.*

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

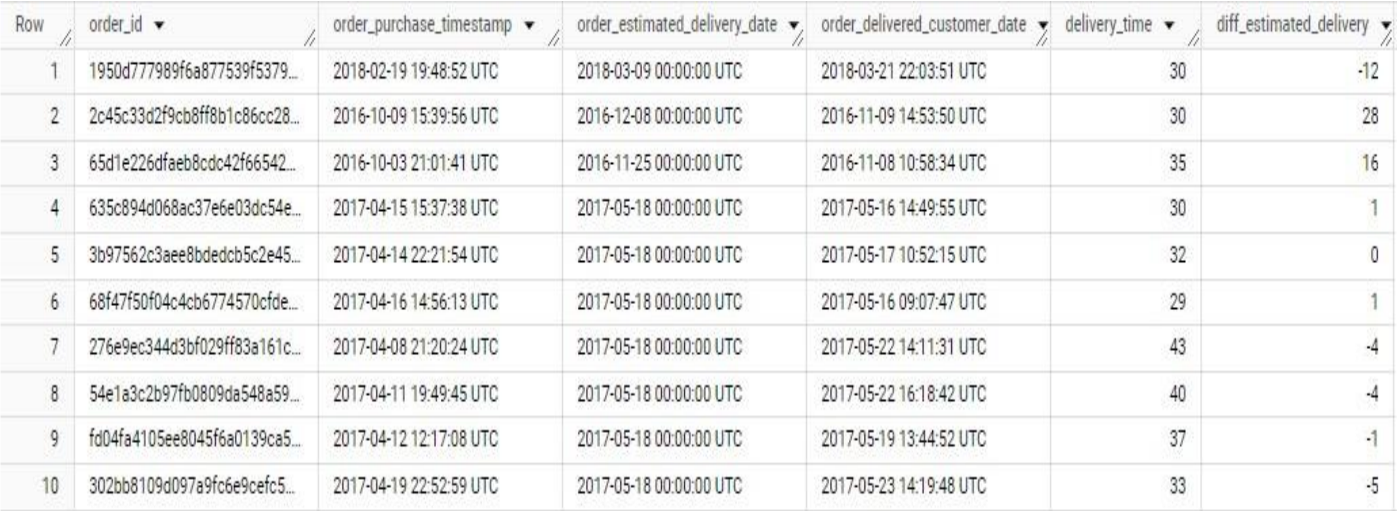
A. Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.

Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

Query->

```
SELECT   order_id,
order_purchase_timestamp,
order_estimated_delivery_date,
order_delivered_customer_date,
  DATE_DIFF( order_delivered_customer_date,
order_purchase_timestamp, day) AS delivery_time,
DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,day) AS diff_estimated_delivery
FROM   target.orders
```

--(In image minus values refer to delay in delivery days)



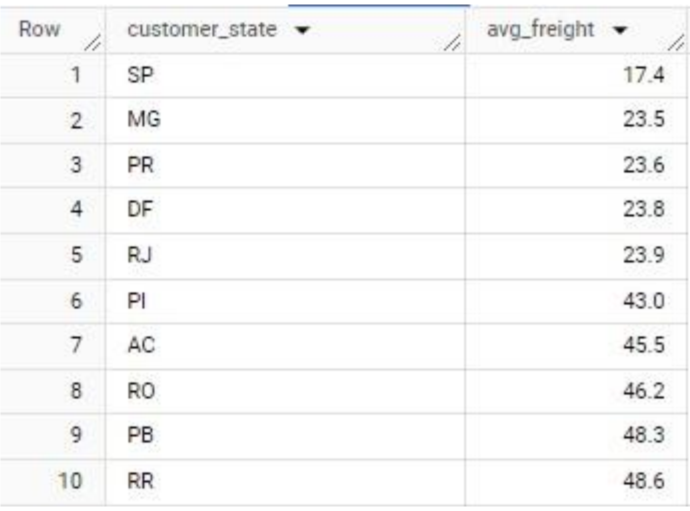
| Row | order_id | order_purchase_timestamp | order_estimated_delivery_date | order_delivered_customer_date | delivery_time | diff_estimated_delivery |
|---|---|---|---|---|---|---|
| 1 | 1950d777989f6a877539f5379... | 2018-02-19 19:48:52 UTC | 2018-03-09 00:00:00 UTC | 2018-03-21 22:03:51 UTC | 30 | -12 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 2016-10-09 15:39:56 UTC | 2016-12-08 00:00:00 UTC | 2016-11-09 14:53:50 UTC | 30 | 28 |
| 3 | 65d1e226dfaeb8cdc42f66542... | 2016-10-03 21:01:41 UTC | 2016-11-25 00:00:00 UTC | 2016-11-08 10:58:34 UTC | 35 | 16 |
| 4 | 635c894d068ac37e6e03dc54e... | 2017-04-15 15:37:38 UTC | 2017-05-18 00:00:00 UTC | 2017-05-16 14:49:55 UTC | 30 | 1 |
| 5 | 3b97562c3aee8bdedcb5c2e45... | 2017-04-14 22:21:54 UTC | 2017-05-18 00:00:00 UTC | 2017-05-17 10:52:15 UTC | 32 | 0 |
| 6 | 68f47f50f04c4cb6774570cfde... | 2017-04-16 14:56:13 UTC | 2017-05-18 00:00:00 UTC | 2017-05-16 09:07:47 UTC | 29 | 1 |
| 7 | 276e9ec344d3bf029ff83a161c... | 2017-04-08 21:20:24 UTC | 2017-05-18 00:00:00 UTC | 2017-05-22 14:11:31 UTC | 43 | -4 |
| 8 | 54e1a3c2b97fb0809da548a59... | 2017-04-11 19:49:45 UTC | 2017-05-18 00:00:00 UTC | 2017-05-22 16:18:42 UTC | 40 | -4 |
| 9 | fd04fa4105ee8045f6a0139ca5... | 2017-04-12 12:17:08 UTC | 2017-05-18 00:00:00 UTC | 2017-05-19 13:44:52 UTC | 37 | -1 |
| 10 | 302bb8109d097a9fc6e9cefc5... | 2017-04-19 22:52:59 UTC | 2017-05-18 00:00:00 UTC | 2017-05-23 14:19:48 UTC | 33 | -5 |

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

Query->

```
        WITH avg_freight_per_state AS (
            SELECT
                c.customer_state,
                ROUND(AVG(order_total_freight),1) AS avg_freight
            FROM (       SELECT
        oi.order_id, SUM(oi.freight_value)
            AS order_total_freight
        FROM
        target.order_items oi
                GROUP BY oi.order_id
        ) order_freight_totals
        JOIN          target.orders
        o
                ON order_freight_totals.order_id
                = o.order_id
        JOIN
        target.customers c
                ON o.customer_id = c.customer_id
            GROUP BY
                c.customer_state
        ) SELECT   customer_state,
        avg_freight
        FROM (  SELECT    customer_state,
        avg_freight,
            ROW_NUMBER() OVER (ORDER BY avg_freight ASC) AS rn_asc,
        ROW_NUMBER() OVER (ORDER BY avg_freight DESC) AS rn_desc
        FROM     avg_freight_per_state
        ) ranked
```

```
 WHERE   rn_asc <= 5 OR rn_desc <= 5  ORDER BY
avg_freight;
```



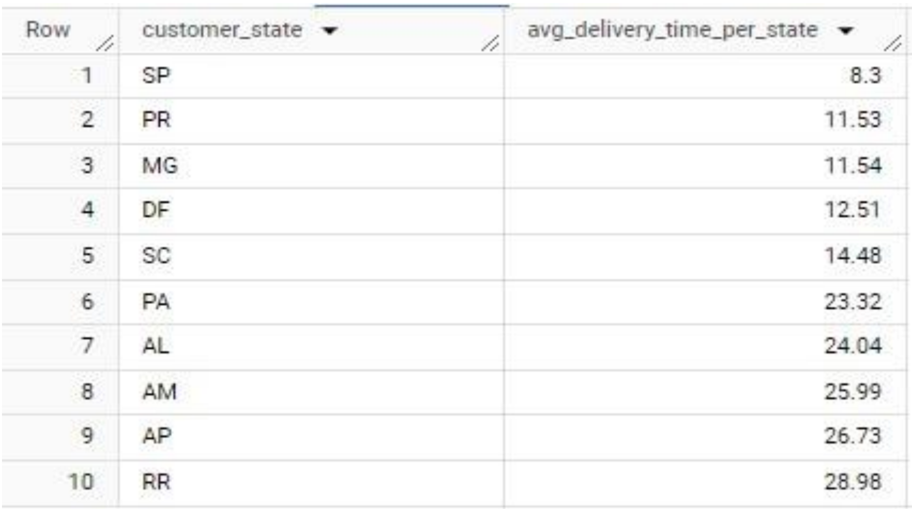
| Row | customer_state | avg_freight |
|---|---|---|
| 1 | SP | 17.4 |
| 2 | MG | 23.5 |
| 3 | PR | 23.6 |
| 4 | DF | 23.8 |
| 5 | RJ | 23.9 |
| 6 | PI | 43.0 |
| 7 | AC | 45.5 |
| 8 | RO | 46.2 |
| 9 | PB | 48.3 |
| 10 | RR | 48.6 |

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

Query->

```
WITH avg_delivery_per_state AS (
     SELECT
       c.customer_state,
      ROUND(AVG(DATE_DIFF( order_delivered_customer_date,
   order_purchase_timestamp, day)),2) AS avg_delivery_time_per_state
     FROM
       `target.orders` o
     JOIN
       `target.customers` c ON o.customer_id = c.customer_id
     GROUP BY
       c.customer_state
     ORDER BY 2 desc
   )
```

```
SELECT
  customer_state,   avg_delivery_time_per_state
FROM (   SELECT
customer_state,
avg_delivery_time_per_state,
    ROW_NUMBER() OVER (ORDER BY avg_delivery_time_per_state ASC) AS
rn_asc,
    ROW_NUMBER() OVER (ORDER BY avg_delivery_time_per_state DESC)
AS rn_desc   FROM
avg_delivery_per_state
) ranked WHERE   rn_asc <= 5
OR rn_desc <= 5
ORDER BY
  2
```

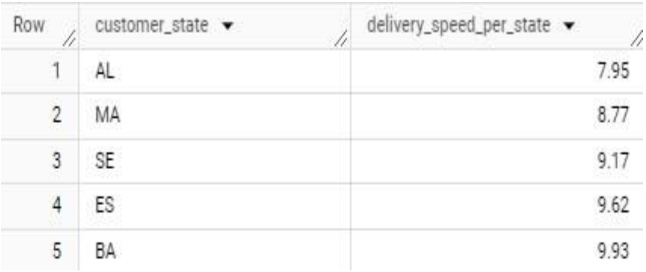


| Row | customer_state | avg_delivery_time_per_state |
|---|---|---|
| 1 | SP | 8.3 |
| 2 | PR | 11.53 |
| 3 | MG | 11.54 |
| 4 | DF | 12.51 |
| 5 | SC | 14.48 |
| 6 | PA | 23.32 |
| 7 | AL | 24.04 |
| 8 | AM | 25.99 |
| 9 | AP | 26.73 |
| 10 | RR | 28.98 |

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

Query –>

```
 SELECT
customer_state,
 ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
     order_delivered_customer_date, day)),2)
AS delivery_speed_per_state FROM   target.orders o
JOIN
  `target.customers` c
ON
  o.customer_id = c.customer_id
WHERE
order_delivered_customer_date
IS NOT NULL GROUP
BY
customer_state
ORDER BY
  2
Limit 5
```



| Row | customer_state | delivery_speed_per_state |
|---|---|---|
| 1 | AL | 7.95 |
| 2 | MA | 8.77 |
| 3 | SE | 9.17 |
| 4 | ES | 9.62 |
| 5 | BA | 9.93 |

## Insights -

- *We can see the delay between Order estimated delivery time and Order delivery time clearly 7 to 19 days varying from state to state.*
- *Maximum delay in order is seen 188 days which is not a good sign for prestige of the company hence delivery after 45 days should be*

*Handled as priority.*

- *Fastest average delivery state wise when compared using estimated delivery can be seen in AL,MA while slowest delivery can be seen in RO and AC*
- *Lowest Average Freight for states can be seen in SP and PR while  Hightest can be seen in RR and AP*

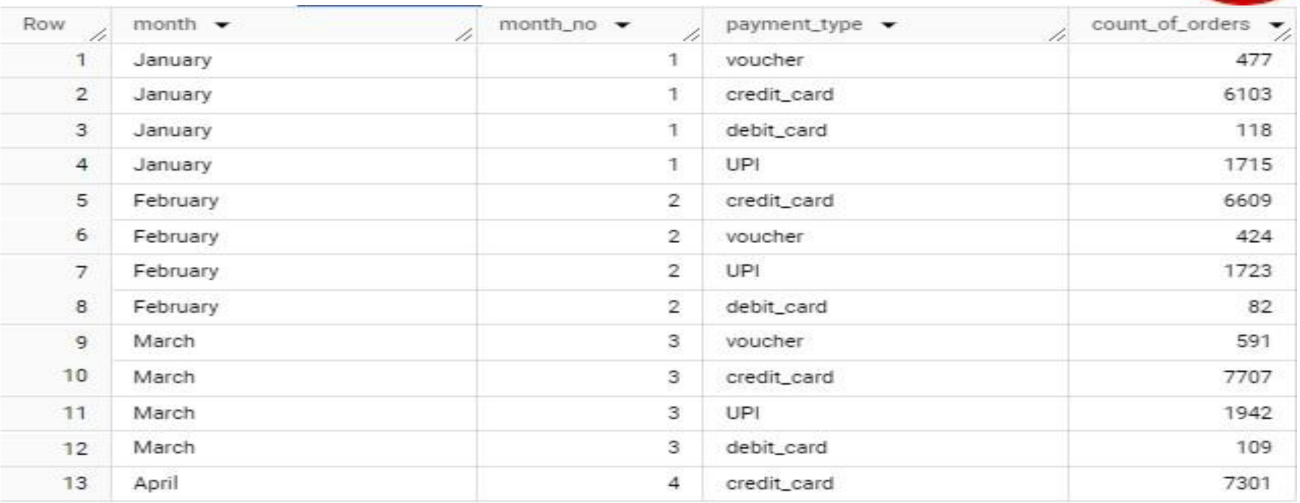
## VI. Analysis based on the payments:

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

Query->

```
SELECT
  FORMAT_DATE('%B',order_purchase_timestamp) AS month,
  EXTRACT(month   FROM
order_purchase_timestamp) AS month_no,
payment_type,
  COUNT(o.order_id) count_of_orders
FROM
  `target.payments` p  JOIN  `target.orders` o
```

```
ON
  o.order_id = p.order_id
GROUP BY
  1,2,3
ORDER BY
  2
```

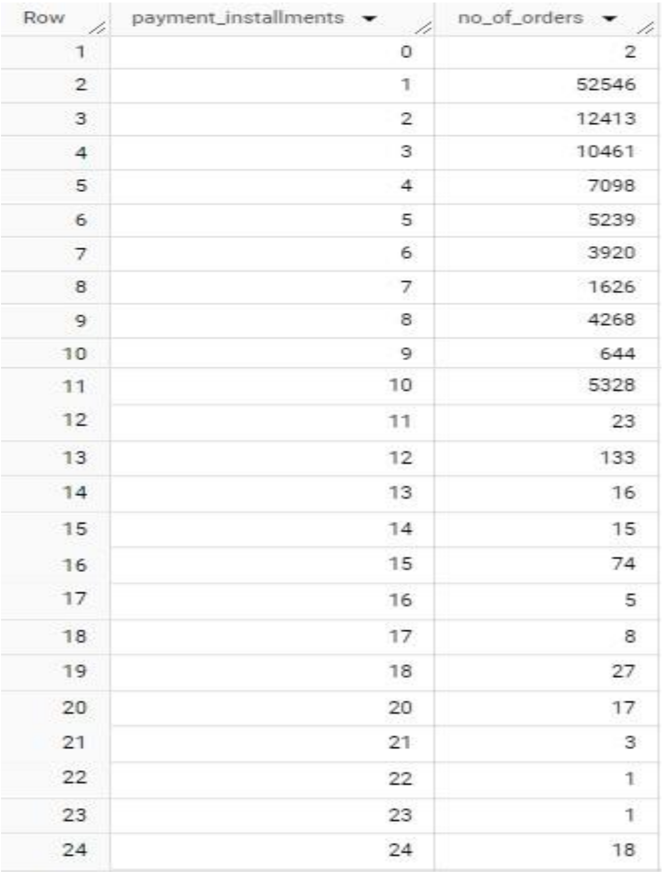


| Row | month | month_no | payment_type | count_of_orders |
|---|---|---|---|---|
| 1 | January | 1 | voucher | 477 |
| 2 | January | 1 | credit_card | 6103 |
| 3 | January | 1 | debit_card | 118 |
| 4 | January | 1 | UPI | 1715 |
| 5 | February | 2 | credit_card | 6609 |
| 6 | February | 2 | voucher | 424 |
| 7 | February | 2 | UPI | 1723 |
| 8 | February | 2 | debit_card | 82 |
| 9 | March | 3 | voucher | 591 |
| 10 | March | 3 | credit_card | 7707 |
| 11 | March | 3 | UPI | 1942 |
| 12 | March | 3 | debit_card | 109 |
| 13 | April | 4 | credit_card | 7301 |

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

Query->

```
 SELECT
payment_installments,
  COUNT(order_id)
no_of_orders FROM
target.payments p WHERE
payment_sequential > 0 GROUP
BY   payment_installments
```



| Row | payment_installments | no_of_orders |
|---|---|---|
| 1 | 0 | 2 |
| 2 | 1 | 52546 |
| 3 | 2 | 12413 |
| 4 | 3 | 10461 |
| 5 | 4 | 7098 |
| 6 | 5 | 5239 |
| 7 | 6 | 3920 |
| 8 | 7 | 1626 |
| 9 | 8 | 4268 |
| 10 | 9 | 644 |
| 11 | 10 | 5328 |
| 12 | 11 | 23 |
| 13 | 12 | 133 |
| 14 | 13 | 16 |
| 15 | 14 | 15 |
| 16 | 15 | 74 |
| 17 | 16 | 5 |
| 18 | 17 | 8 |
| 19 | 18 | 27 |
| 20 | 20 | 17 |
| 21 | 21 | 3 |
| 22 | 22 | 1 |
| 23 | 23 | 1 |
| 24 | 24 | 18 |

## Insights -

- *As we can see from data Maximum Payments have been made through Credit Card and Minimum Payments are made through Debit Card.*
- *Maximum Orders are paid in one and two installments also, count of orders decreased with the the increase in number of payments*

## Recommendations Analyzing overall Business case –

- *Boost Marketing During Busy Times**:** Increase promotions during high-sales periods like year-end festivals and Black Friday to maximize sales when customers are most active.*

- *Improve Afternoon Services: Since most sales happen in the afternoon, assign more staff and resources during these hours to better serve customers and handle more orders.*

- *Expand in Low-Sales Regions: Launch targeted marketing campaigns in states like Roraima (RR), Amapa (AP), and Acre (AC) to attract more customers where sales are currently low.*

- *Speed Up Deliveries: Work on reducing delivery delays, especially in areas where orders take too long. Make sure no delivery takes more than 45 days to keep customers happy.*

- *Encourage Bigger Orders: Offer discounts or free shipping for larger purchases to motivate customers to buy more and help reduce shipping costs per order.*

- *Simplify Payment Options: Make paying easier by promoting popular methods like credit cards and offering deals on other payment types. Provide simple installment plans to help customers make bigger purchases without hassle.*

- *Focus on Strong Markets: Create loyalty programs in states with many customers, like Sao Paulo (SP), to keep them coming back and increase repeat sales.*

- *Adjust Operations to Demand: Expand your operations during times when sales grow a lot, like in January, so you can meet customer demand and not miss out on potential sales.*