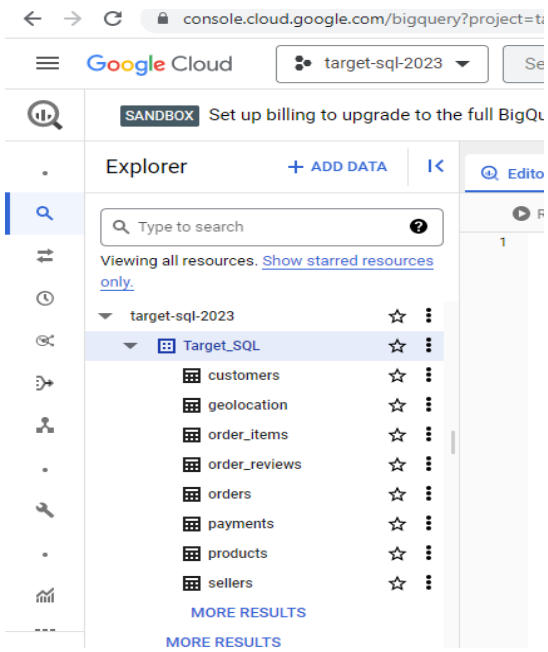
# TARGET SQL PROJECT

Name : Apoorva Jadhav
Batch : DSML AUG 2022 BEGINNER MORNING

## 1)Import the dataset



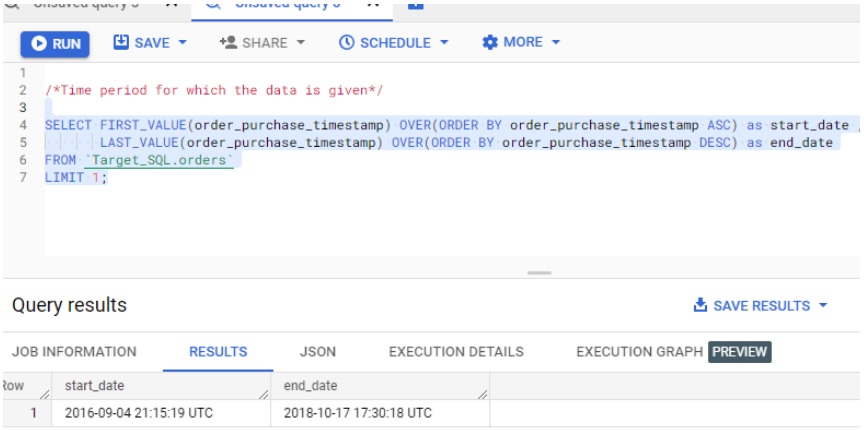

Data set imported successfully from the given drive under the project Target-sql-2023

### 1.1)Data type of columns in a table

| 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 |

The above table shows the datatype of columns in customer table

1.2)Time Period for which the data is given



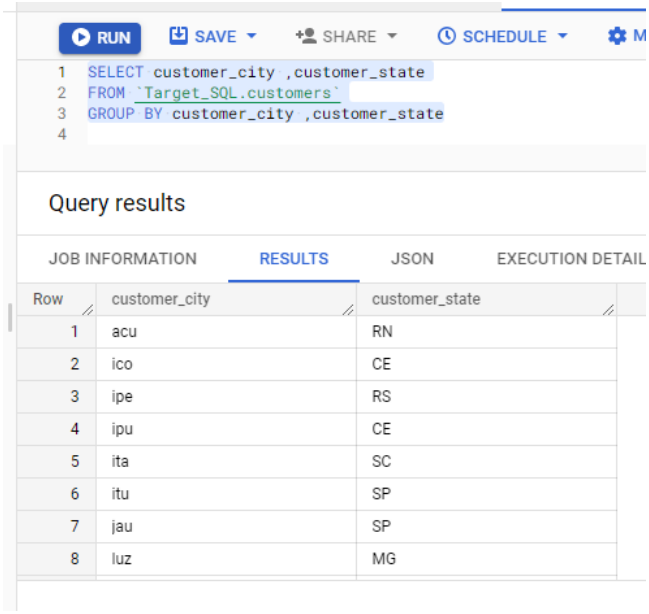
```
/*Time period for which the data is given*/

SELECT FIRST_VALUE(order_purchase_timestamp) OVER(ORDER BY order_purchase_timestamp ASC) as start_date
       LAST_VALUE(order_purchase_timestamp) OVER(ORDER BY order_purchase_timestamp DESC) as end_date
FROM `Target_SQL.orders`
LIMIT 1;
```

Query results

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

1.3)Cities and States of customers ordered during the given period

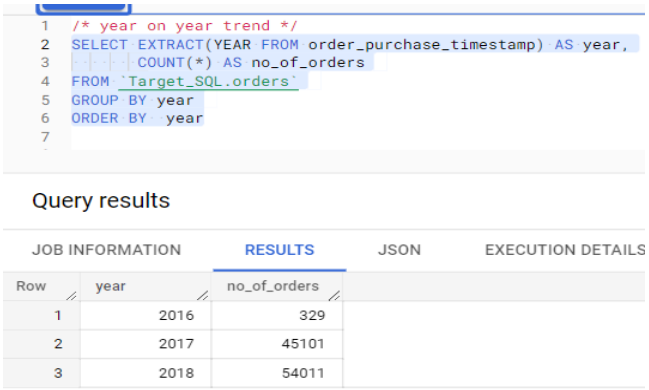


```
SELECT customer_city ,customer_state
FROM `Target_SQL.customers`
GROUP BY customer_city ,customer_state
```

Query results

| Row | customer_city | customer_state |
|---|---|---|
| 1 | acu | RN |
| 2 | ico | CE |
| 3 | ipe | RS |
| 4 | ipu | CE |
| 5 | ita | SC |
| 6 | itu | SP |
| 7 | jau | SP |
| 8 | luz | MG |

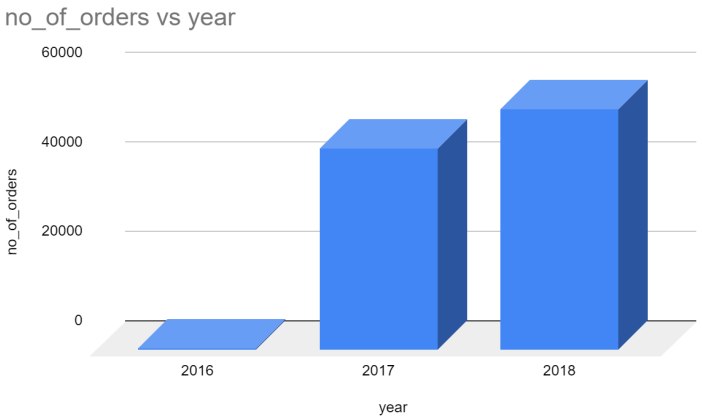
**2)In-depth Exploration:**Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? 2.1)Year-on-Year analysis



```
/* year on year trend */
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
       COUNT(*) AS no_of_orders
FROM `Target_SQL.orders`
GROUP BY year
ORDER BY year
```

Query results

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




As we can see from the chart, the no of orders have been increasing proportionally with year.Hence we can say that there is a growing trend in ecommerce with years

## 2.2)Can we see some seasonality with peaks at specific months?

RUN SAVE SHARE SCHEDULE MOR

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
       COUNT(*) AS no_of_orders
FROM `Target_SQL.orders`
GROUP BY year , month
ORDER BY year , month
```

JOB INFORMATION | RESULTS | JSON | EXE

| Row | year | month | no_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 |

We can see that there is a spike in orders during the October – November 2017 and there is a sharp decline in Orders during the August – September – October 2018 period. Declining during November – December period.

2.2)What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)

RUN SAVE SHARE SCHEDULE MORE

```sql
with orders_per_hour AS
(
  SELECT  EXTRACT(HOUR FROM order_purchase_timestamp) AS hour,
          COUNT(*) AS no_of_orders
  FROM `Target_SQL.orders`
  GROUP BY hour

)
SELECT  SUM(no_of_orders) AS total_orders,
        CASE
        WHEN hour BETWEEN 1 AND 6 THEN 'dawn'
        WHEN hour BETWEEN 5 AND 13 THEN 'morning'
        WHEN hour BETWEEN 12 AND 19 THEN 'afternoon'
        ELSE 'night'
        END AS timings
FROM orders_per_hour
GROUP BY timings
ORDER BY total_orders
```

JOB INFORMATION | RESULTS | JSON | EXECUTIO

| Row | total_orders | timings |
|---|---|---|
| 1 | 2848 | dawn |
| 2 | 24743 | night |
| 3 | 34251 | morning |
| 4 | 37599 | afternoon |

According to the above data , customers tend to order more during afternoon i.e between 12 pm and 18pm .The least orders in dawn . Orders are comparatively less in the night than morning or afternoon.

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

3.1)Get month on month orders by states

```sql
SELECT c.customer_state as state,
       EXTRACT (MONTH FROM o.order_purchase_timestamp) AS month,
       count(o.order_id) AS no_of_orders
FROM `Target_SQL.orders` AS o
LEFT JOIN `Target_SQL.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY state,month
ORDER BY state,month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

| Row | state | month | no_of_orders |
|---|---|---|---|
| 1 | AC | 1 | 8 |
| 2 | AC | 2 | 6 |
| 3 | AC | 3 | 4 |
| 4 | AC | 4 | 9 |
| 5 | AC | 5 | 10 |
| 6 | AC | 6 | 7 |
| 7 | AC | 7 | 9 |
| 8 | AC | 8 | 7 |
| 9 | AC | 9 | 5 |
| 10 | AC | 10 | 6 |
| 11 | AC | 11 | 5 |
| 12 | AC | 12 | 5 |
| 13 | AL | 1 | 39 |
| 14 | AL | 2 | 39 |

Load more

The above table gives the details of month on month orders of each customer state.

## 3.2)Distribution of customers across the states in Brazil

RUN | SAVE | SHARE | SCHEDULE | MORE

```
1  SELECT customer_state , customer_city,
2         count(customer_id) AS no_of_customers
3  FROM `Target_SQL.customers`
4  GROUP BY customer_state,customer_city
5  ORDER BY no_of_customers desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

| Row | customer_state | customer_city | no_of_customer |
|---|---|---|---|
| 1 | SP | sao paulo | 15540 |
| 2 | RJ | rio de janeiro | 6882 |
| 3 | MG | belo horizonte | 2773 |
| 4 | DF | brasilia | 2131 |
| 5 | PR | curitiba | 1521 |
| 6 | SP | campinas | 1444 |
| 7 | RS | porto alegre | 1379 |
| 8 | BA | salvador | 1245 |
| 9 | SP | guarulhos | 1189 |
| 10 | SP | sao bernardo do campo | 938 |

Result

Total Records are  4310 which cannot be shown on a single graph.

Hence lets try distribution of customers by each state to get the overall distribution of customers

RUN SAVE SHARE SCHEDULE

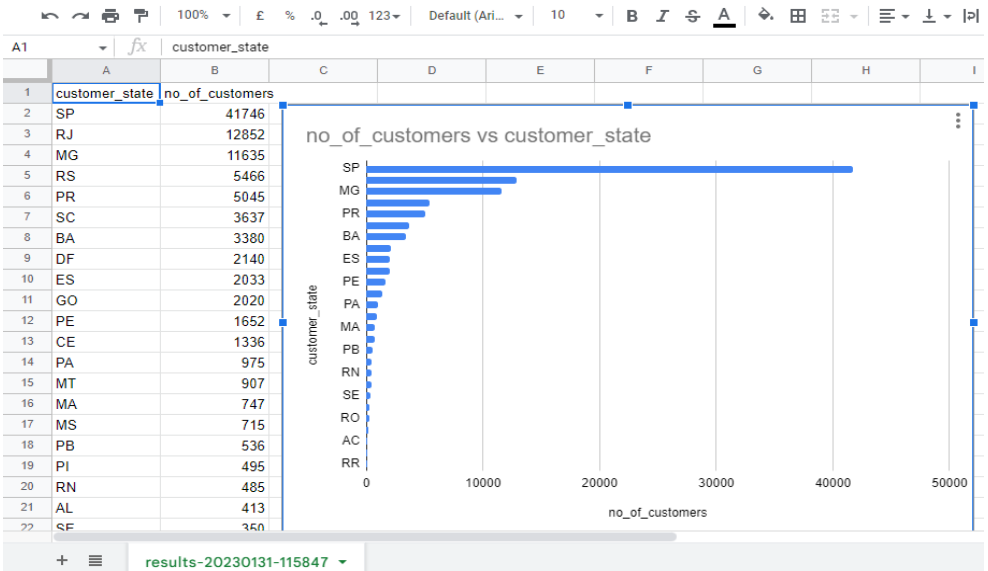
```
SELECT customer_state ,
       count(customer_id) AS no_of_customers
FROM `Target_SQL.customers`
GROUP BY customer_state
ORDER BY no_of_customers desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUT

| Row | customer_state | no_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 |

The below graph shows the top 3 states having maximum customers
I.e SP , RJ and MG respectively. State RR has the least number of customers i.e 46



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

4.1)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

```sql
WITH p1 AS
(
  SELECT payment_value,
         EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
         EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  FROM `Target_SQL.orders` AS o
  JOIN `Target_SQL.payments` AS p
  ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 and
        EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 and 8
) ,
p2 AS
(SELECT payment_value,
         EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
         EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  FROM `Target_SQL.orders` AS o
  JOIN `Target_SQL.payments` AS p
  ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM order_purchase_timestamp) = 2018 and
        EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 and 8
)
SELECT round(((sum(p2.payment_value) - sum(p1.payment_value))/ sum(p1.payment_value)) *100 , 3) AS
       percentage_cost_increase
FROM p1 JOIN p2
ON p1.month = p2.month
```

Query results

JOB INFORMATION | RESULTS

| Row | percentage_cost_increase |
|---|---|
| 1 | 4.211 |

4.2)Mean & Sum of price and freight value by customer state

RUN | SAVE | SHARE | SCHEDULE | MORE

```sql
SELECT c.customer_state,
       AVG(o1.price) AS mean_price,
       SUM(o1.price) AS total_price,
       AVG(o1.freight_value) AS mean_freight,
       SUM(o1.freight_value) AS total_freight,
       AVG(o1.price + o1.freight_value) AS mean_cost_of_order,
       SUM(o1.price + o1.freight_value) AS total_cost_of_order,
FROM `Target_SQL.order_items` as o1
JOIN `Target_SQL.orders` as o2
ON o1.order_id = o2.order_id
JOIN `Target_SQL.customers` AS c
ON o2.customer_id = c.customer_id
GROUP BY c.customer_state
```

Query results

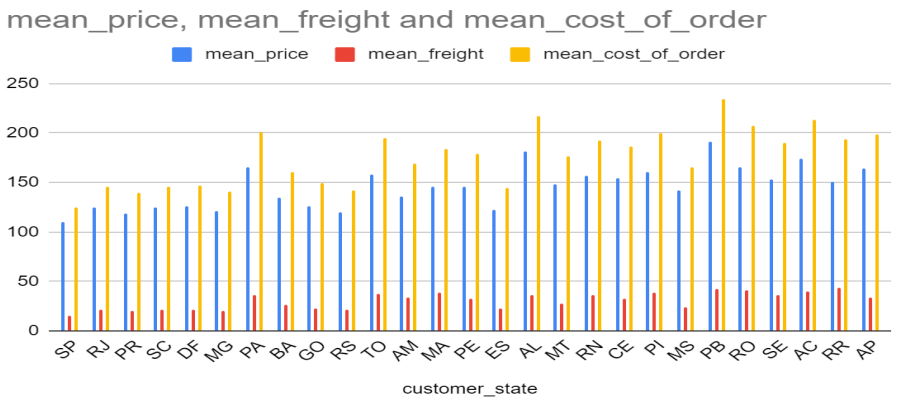
SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

| Row | customer_state | mean_price | total_price | mean_freight | total_freight | mean_cost_of_o | total_cost_of_or |
|---|---|---|---|---|---|---|---|
| 1 | SP | 109.65362915972931 | 5202955.05... | 15.1472753... | 718723.069... | 124.800904... | 5921678.11... |
| 2 | RJ | 125.11781809451907 | 1824092.66... | 20.9609239... | 305589.310... | 146.078742... | 2129681.97... |
| 3 | PR | 119.00413937282218 | 683083.760... | 20.5316515... | 117851.680... | 139.535790... | 800935.440... |
| 4 | SC | 124.65357758620696 | 520553.340... | 21.4703687... | 89660.2600... | 146.123946... | 610213.600... |
| 5 | DF | 125.77054862842866 | 302603.939... | 21.0413549... | 50625.4999... | 146.811903... | 353229.440... |
| 6 | MG | 120.74857414883108 | 1585308.02... | 20.6301668... | 270853.460... | 141.378740... | 1856161.48... |
| 7 | PA | 165.69241666666659 | 178947.809... | 35.8326851... | 38699.3000... | 201.525101... | 217647.109... |
| 8 | BA | 134.60120821268725 | 511349.990... | 26.3639589... | 100156.679... | 160.965167... | 611506.670... |
| 9 | GO | 126.27173167595375 | 294591.949... | 22.7668152... | 53114.9799... | 149.038546... | 347706.930... |
| 10 | RS | 120.33745308741014 | 750304.020... | 21.7358043... | 135522.740... | 142.073257... | 885826.760... |
| 11 | TO | 157.52933333333331 | 49621.7400... | 37.2466031... | 11732.6799... | 194.775936... | 61354.4200... |
| 12 | AM | 135.49599999999998 | 22356.8400... | 33.2053939... | 5478.89000... | 168.701393... | 27835.7299... |
| 13 | MA | 145.20415048543708 | 119648.219... | 38.2570024... | 31523.7700... | 183.461152... | 151171.989... |

Results per page: 50 | 1 – 27 of 27

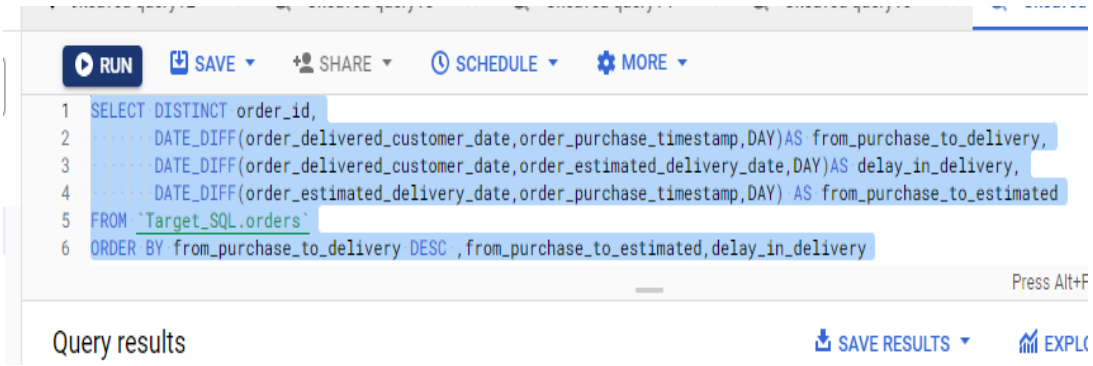
The output is more readable through graph below




State PB has the highest mean_price, mean_freight and mean_cost_of_order and State SP being the lowest of all.

## 5)Analysis on sales, freight and delivery time

5.1)Calculate days between purchasing, delivering and estimated delivery



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

| Row | order_id | from_purchase_to_delivery | delay_in_delivery | from_purchase_to_estimated |
|---|---|---|---|---|
| 1 | ca07593549f1816d26a5... | 209 | 181 | 28 |
| 2 | 1b3190b2dfa9d789e1f1... | 208 | 188 | 19 |
| 3 | 440d0d17af552815d15a... | 195 | 165 | 30 |
| 4 | 285ab9426d698203452... | 194 | 166 | 28 |
| 5 | 0f4519c5f1c541ddec9f2... | 194 | 161 | 32 |
| 6 | 2fb597c2f772eca01b1f5... | 194 | 155 | 39 |
| 7 | 47b40429ed8cce3aee91... | 191 | 175 | 15 |
| 8 | 2fe324febf907e3ea3f2a... | 189 | 167 | 22 |
| 9 | 2d7561026d542c8dbd8f... | 188 | 159 | 28 |
| 10 | c27815f7e3dd0b926b58... | 187 | 162 | 25 |
| 11 | 437222e3fd1b07396f1d... | 187 | 144 | 42 |
| 12 | dfe5f68118c257614324... | 186 | 153 | 32 |
| 13 | 6e82dcfb5eada6283dba... | 182 | 155 | 27 |

Load more

Results per page: 50 1 – 50 of 99441

There are a total of 99441 orders placed in the data set. We can see the delay in the delivery for the orders placed.Some of the reasons for shipment delay are

- Extreme weather conditions like draughts,landslide,heavy rains,etc
- Spike in orders during holidays
- Inaccurate shipping details. As there are some null values in order_delivery_date or order_estimated_delivery_date

5.2)Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

RUN SAVE SHARE SCHEDULE MORE

```
SELECT DISTINCT order_id,
       DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)AS time_to_delivery,
       DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY) AS diff_estimated_delivery
FROM `Target_SQL.orders`
ORDER BY time_to_delivery desc,diff_estimated_delivery
```

Press Alt+F

Query results SAVE RESULTS EXPL

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

| Row | order_id | time_to_delivery | diff_estimated_d |
|---|---|---|---|
| 1 | ca07593549f1816d26a572e06... | 209 | 181 |
| 2 | 1b3190b2dfa9d789e1f14c05b... | 208 | 188 |
| 3 | 440d0d17af552815d15a9e41a... | 195 | 165 |
| 4 | 2fb597c2f772eca01b1f5c561b... | 194 | 155 |
| 5 | 0f4519c5f1c541ddec9f21b3bd... | 194 | 161 |
| 6 | 285ab9426d6982034523a855f... | 194 | 166 |
| 7 | 47b40429ed8cce3aee9199792... | 191 | 175 |
| 8 | 2fe324febf907e3ea3f2aa9650... | 189 | 167 |
| 9 | 2d7561026d542c8dbd8f0daea... | 188 | 159 |
| 10 | 437222e3fd1b07396f1d9ba8c... | 187 | 144 |

Results per page: 50 1 – 50 of 99441

As we see from the above table first order there is seen a delay of 28 days between time of estimated delivery to actual delivery.

## 5.3)Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

aved query13 *Unsaved query14 *Unsaved query15 *Unsaved query16 *Unsaved query1

RUN SAVE SHARE SCHEDULE MORE Query complete

```
SELECT customer_state,
       ROUND(AVG(freight_value) , 2) AS mean_freight_value,
       ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)))AS mean_time_to_delivery,
       ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY))) AS mean_diff_estimated_delivery
FROM `Target_SQL.order_items` AS o1
JOIN `Target_SQL.orders` as o2
ON o1.order_id = o2.order_id
JOIN `Target_SQL.customers` as c
ON c.customer_id = o2.customer_id
GROUP BY c.customer_state
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

| Row | customer_state | mean_freight_value | mean_time_to_delivery | mean_diff_estimated_delivery |
|---|---|---|---|---|
| 1 | MT | 28.17 | 18.0 | 14.0 |
| 2 | MA | 38.26 | 21.0 | 9.0 |
| 3 | AL | 35.84 | 24.0 | 8.0 |
| 4 | SP | 15.15 | 8.0 | 10.0 |
| 5 | MG | 20.63 | 12.0 | 12.0 |
| 6 | PE | 32.92 | 18.0 | 13.0 |
| 7 | RJ | 20.96 | 15.0 | 11.0 |
| 8 | DF | 21.04 | 13.0 | 11.0 |
| 9 | RS | 21.74 | 15.0 | 13.0 |
| 10 | SE | 36.65 | 21.0 | 9.0 |

5.4)Sort the data to get the following:

- Top 5 states with lowest average freight value - sorted in asc limit 5

RUN SAVE SHARE SCHEDULE

```
SELECT customer_state,
       ROUND(AVG(freight_value),2) AS mean_freight_value
FROM `Target_SQL.order_items` AS o1
JOIN `Target_SQL.orders` as o2
ON o1.order_id = o2.order_id
JOIN `Target_SQL.customers` as c
ON c.customer_id = o2.customer_id
GROUP BY c.customer_state
ORDER BY mean_freight_value
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DE

| Row | customer_state | mean_freight_va |
|---|---|---|
| 1 | SP | 15.15 |
| 2 | PR | 20.53 |
| 3 | MG | 20.63 |
| 4 | RJ | 20.96 |
| 5 | DF | 21.04 |

- Top 5 states with highest average time to delivery

RUN SAVE SHARE SCHEDULE MORE

```
SELECT customer_state,
       ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)))
       AS avg_time_to_delivery,
FROM `Target_SQL.order_items` AS o1
JOIN `Target_SQL.orders` as o2
ON o1.order_id = o2.order_id
JOIN `Target_SQL.customers` as c
ON c.customer_id = o2.customer_id
GROUP BY c.customer_state
ORDER BY avg_time_to_delivery DESC
```

Query results SAVE RE

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREV

| Row | customer_state | avg_time_to_delivery |
|---|---|---|
| 1 | AP | 28.0 |
| 2 | RR | 28.0 |
| 3 | AM | 26.0 |
| 4 | AL | 24.0 |
| 5 | PA | 23.0 |

avg_time_to_delivery

State AP has 28.0 , the highest average time to delivery followed by RR and AM.

- Top 5 states where delivery is not so fast compared to estimated date

  (higher the difference - so is the delay in delivery)

RUN SAVE SHARE SCHEDULE MORE

```
SELECT customer_state,
       ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)))
       AS mean_diff_estimated_delivery
FROM `Target_SQL.order_items` AS o1
JOIN `Target_SQL.orders` as o2
ON o1.order_id = o2.order_id
JOIN `Target_SQL.customers` as c
ON c.customer_id = o2.customer_id
GROUP BY c.customer_state
ORDER BY mean_diff_estimated_delivery DESC
LIMIT 5
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

| Row | customer_state | mean_diff_estim |
|---|---|---|
| 1 | AC | 20.0 |
| 2 | AM | 19.0 |
| 3 | RO | 19.0 |
| 4 | RR | 17.0 |
| 5 | AP | 17.0 |

## 6) Payment type analysis

6.1)Month over Month count of orders for different payment types

RUN SAVE SHARE SCHEDULE MORE

```
SELECT  CONCAT(EXTRACT(YEAR FROM order_purchase_timestamp),'-',
        EXTRACT(MONTH FROM order_purchase_timestamp)) AS year_month,
        p.payment_type,
        COUNT(o.order_id) AS no_of_orders
FROM `Target_SQL.orders` AS o
JOIN `Target_SQL.payments` AS p
ON o.order_id = p.order_id
GROUP BY year_month ,p.payment_type
ORDER BY year_month
```

JOB INFORMATION | RESULTS | JSON

| Row | year_month | payment_type | no_of_orders |
|---|---|---|---|
| 1 | 2016-10 | credit_card | 254 |
| 2 | 2016-10 | UPI | 63 |
| 3 | 2016-10 | voucher | 23 |
| 4 | 2016-10 | debit_card | 2 |
| 5 | 2016-12 | credit_card | 1 |
| 6 | 2016-9 | credit_card | 3 |
| 7 | 2017-1 | credit_card | 583 |
| 8 | 2017-1 | UPI | 197 |
| 9 | 2017-1 | voucher | 61 |
| 10 | 2017-1 | debit_card | 9 |

6.2)Count of orders based on the no. of payment instalments

```
RUN   SAVE   SHARE
1  SELECT payment_installments,
2         count(order_id) AS no_of_orders
3  FROM `Target_SQL.payments`
4  GROUP BY payment_installments
5  ORDER BY no_of_orders desc
```

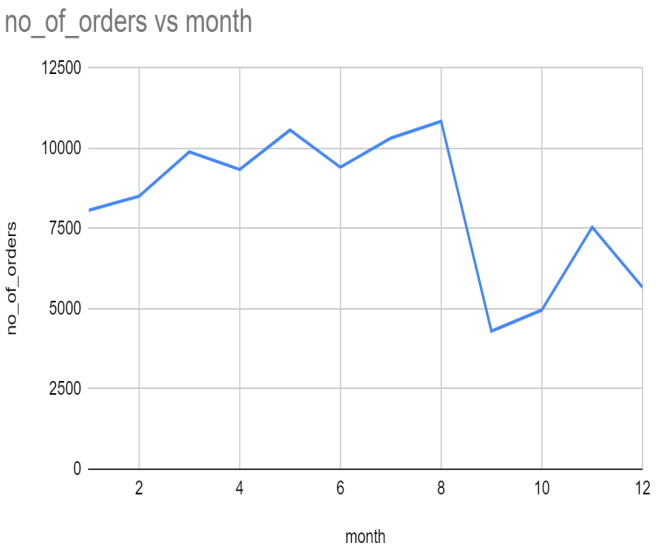
Query results

JOB INFORMATION | RESULTS | JSON

| ow | payment_installments | no_of_orders |
|---|---|---|
| 1 | 1 | 52546 |
| 2 | 2 | 12413 |
| 3 | 3 | 10461 |
| 4 | 4 | 7098 |
| 5 | 10 | 5328 |
| 6 | 5 | 5239 |
| 7 | 8 | 4268 |
| 8 | 6 | 3920 |
| 9 | 7 | 1626 |
| 10 | 9 | 644 |

## 7)Actionable Insights

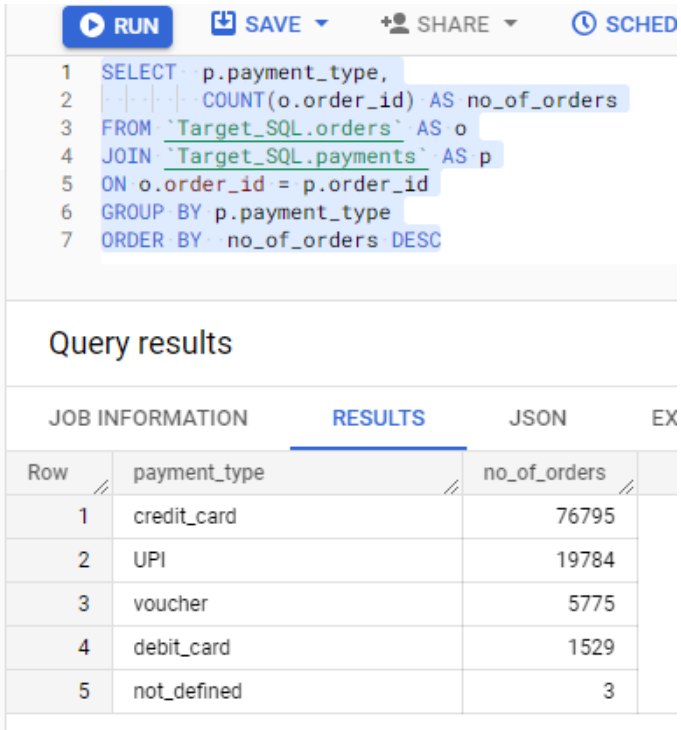
- Time period for the given Target Dataset is between 2016-09-04 21:15:19 UTC and 2018-10-17 17:30:18 UTC
- there is a growing trend in ecommerce i.e from the year 2016 to 2018,the no of orders have been increased showing a good growth to the company
- When we look for the seasonality peaks in month irrespective of year , we can see that there has been steep slope of order in month of September where as month August shows the highest no of orders followed by July and May.



| Row | month | no_of_orders |
|---|---|---|
| 1 | 1 | 8069 |
| 2 | 2 | 8508 |
| 3 | 3 | 9893 |
| 4 | 4 | 9343 |
| 5 | 5 | 10573 |
| 6 | 6 | 9412 |
| 7 | 7 | 10318 |
| 8 | 8 | 10843 |
| 9 | 9 | 4305 |
| 10 | 10 | 4959 |
| 11 | 11 | 7544 |
| 12 | 12 | 5674 |

- The top 3 states having maximum customers are SP , RJ and MG respectively. State RR has the least number of customers i.e 46
- There are total of 27 states included in this Dataset
- We can see the delay in the delivery for the orders placed. From query 5.1
  Some of the reasons for shipment delay may be
    - Extreme weather conditions like draughts,landslide,heavy rains,etc
    - Spike in orders during holidays
    - Inaccurate shipping details. As there are some null values in order_delivery_date or order_estimated_delivery_date.

- <u>Payment done through the credit cards is seen more.</u>



- 5 Cheapest products ordered

RUN SAVE SHARE SCHEDULE MORE

```
SELECT  o1.order_id,p.product_id ,
        p.product_category,
        ( o.price+o.freight_value) as total_price
FROM `Target_SQL.orders` o1
JOIN `Target_SQL.order_items` as o
ON o.order_id = o1.order_id
JOIN `Target_SQL.products` as p
ON p.product_id = o.product_id
GROUP BY o1.order_id , p.product_id,p.product_category,total_price
ORDER BY total_price
LIMIT 5
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

| Row | order_id | product_id | product_category | total_price |
|---|---|---|---|---|
| 1 | 8fbcb92faf1aa60361f61ed7ae... | adc48fd26eea311ca6856b58d... | housewares | 6.08 |
| 2 | 71e22e2d99081d6dc07d9627... | ac480ada5f06d9024eae2c6dd... | perfumery | 7.28 |
| 3 | 8272b63d03f5f79c56e9e4120... | 05b515fdc76e888aada3c6d66... | HEALTH BEAUTY | 9.09 |
| 4 | 8272b63d03f5f79c56e9e4120... | 270516a3f41dc035aa87d2202... | HEALTH BEAUTY | 9.09 |
| 5 | 54f79e394bb60e8e7d5d56756... | a25583531530c0913ea4dee2c... | perfumery | 9.27 |

## 8)Recommendation

- Ways to encounter the delay issues of the shipment :
  - Ensure overstock of products,
  - Help in forecast demand and plan their inventory ,
  - Brand trust is ensured
- Date of birth is to be added to customer table,which can be used to give them gift coupons or discount code on their birthday week or month
- Preference of frequent orders can be used to improvise sales in states performing low on orders
- Review based analysis should be done
- Provide local pickup facility, this can reduce shipping cost as well as time
- Knowing the customer preference of category of products will help in better recommendation and it can make the brand more customer friendly