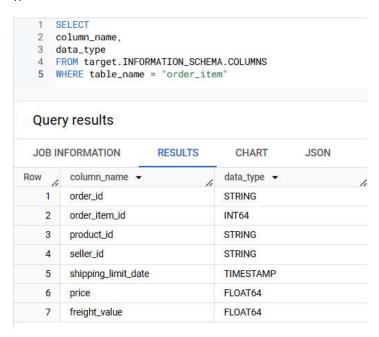
Target Business Case Study Using SQL

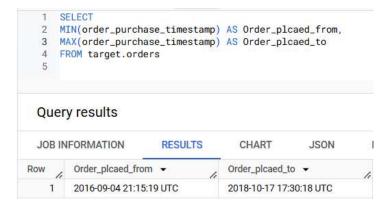
25 February 2025 19:03

Basic data exploration

1) Data type of all Columns in Order Item Table



2) The Time range between the orders are placed



3) Total Number of Unique customers



4) Total number of products and different product category the inventory holds



5) Finding out total number of cities and states in our dataset



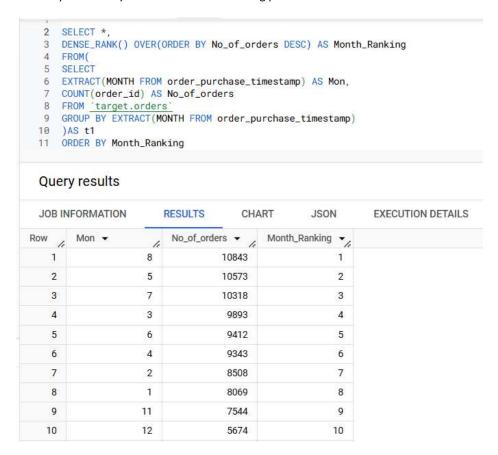
Exploratory Data Analysis

1) Finding growth trend in Number of orders placed over the years



Insight: From the above Trend we can see that there is Significant boost in Orders Placed between 2016 and 2017

2) Is there a monthly Seasonality on number of Orders being place



Insight: From the above Output Maximum Average orders are placed Between May through August

3) During what time of the day customer place most of their Order. Considering below time frames

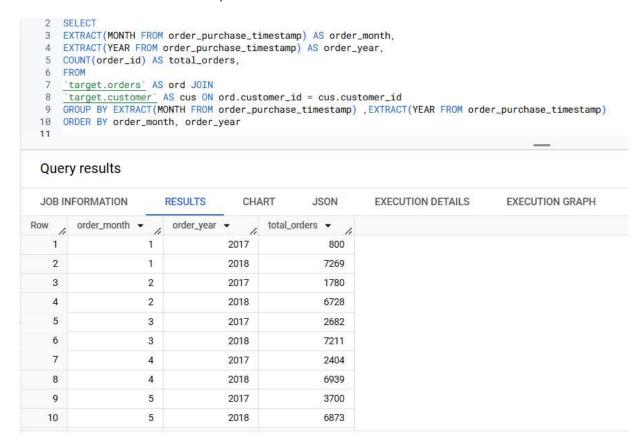
0-6 hrs : Dawn 7-12 hrs : Mornings 13-18 hrs : Afternoon 19-23 hrs : Night

```
1 SELECT
2 Timing,
3 COUNT(order_id) AS Total_Order
4 FROM
5
```

```
SELECT
  2
     Timing,
  3
     COUNT(order_id) AS Total_Order
  4
    FROM
  5
  6
     SELECT *,
  7
     CASE
  8 WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN "Dawn"
  9
     WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN "Morning"
 10
     WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN "Afternoon"
 11 ELSE "Night" END AS Timing
 12 FROM 'target.orders') AS t1
 13 GROUP BY timing
    ORDER BY Total_Order DESC
 14
 15
 Query results
 JOB INFORMATION
                        RESULTS
                                      CHART
                                                   JSON
                                                              EXECUTION DETAILS
                                                                                      EXECUTION GRAPH
Row
        Timing ▼
                                    Total_Order ▼
                                             38135
   1
        Afternoon
   2
        Night
                                             28331
   3
        Morning
                                             27733
   4
        Dawn
                                              5242
```

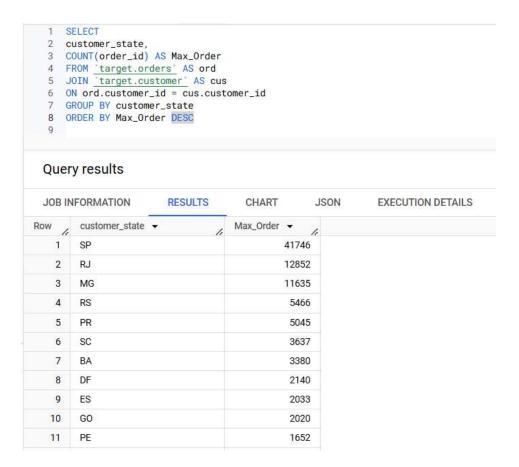
Insight: The output give clear picture that most of the Orders are Placed during Mid of the day

4) Month on Month number of orders Placed over year



Insight: Comparing to previous Year Sales we can see there is Definitely a growth trend in Sales on Each month

5) How are the customer distributed across the state



6) Analysing % of increase in cost of order for 2018 comparing to 2017

```
SELECT
2
    CONCAT(ROUND(((Y_2018 - Y_2017)/Y_2017)*100,2), "% Increase Over 2017 & 2018") AS Increase_Percent
3
    FROM(
4
     MAX(CASE WHEN ord_year = 2017 THEN ROUND(Total_pay_value, 0) END) AS Y_2017,
5
     MAX(CASE WHEN ord_year = 2018 THEN ROUND(Total_pay_value,0) END) AS Y_2018
7
    FROM
8
9
    SELECT
10
     EXTRACT(YEAR FROM order_purchase_timestamp) AS ord_year,
     SUM(payment_value) AS Total_pay_value
11
12
    FROM 'target.orders' AS ord
    JOIN <u>'target.payments'</u> AS pay
13
14
    ON ord.order_id = pay.order_id
    GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp)) AS t1)AS t2
15
16
Query results
JOB INFORMATION
                      RESULTS
                                    CHART
                                                 JSON
                                                                                    EXECUTION GRAPH
                                                            EXECUTION DETAILS
      Increase_Percent ▼
  1
      20% Increase Over 2017 & 2018
```

7) Calculating total & average value of order cost amount of each state

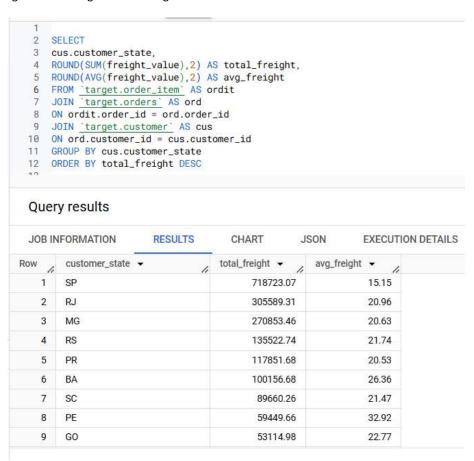
```
2 cus.customer_state,
  3 ROUND(SUM(payment_value),2) AS Total_order_price,
  4 ROUND(AVG(payment_value), ∅) AS Average_order_price
  6 FROM <u>'target.orders'</u> AS ord
  7 JOIN 'target.customer'AS cus
  8 ON ord.customer_id = cus.customer_id
     JOIN <u>'target.payments'</u> AS pay
 10 ON pay.order_id = ord.order_id
 11 GROUP BY cus.customer_state
 12 ORDER BY Total_order_price DESC
 Query results
 JOB INFORMATION
                                                    JSON
                         RESULTS
                                       CHART
                                                                EXECUTION DETAILS
                                     Total_order_price >
        customer_state -
                                                      Average_order_price
Row
   1
        SP
                                          5998226.96
                                                                 138.0
   2
        RJ
                                          2144379.69
                                                                 159.0
                                                                 155.0
   3
        MG
                                          1872257.26
   4
                                           890898.54
                                                                 157.0
        RS
   5
        PR
                                           811156.38
                                                                 154.0
   6
        SC
                                           623086.43
                                                                 166.0
   7
                                                                 171.0
        BA
                                           616645.82
   8
        DF
                                           355141.08
                                                                 161.0
   9
        GO
                                           350092.31
                                                                 166.0
```

8) Calculating total & average value of order price of each state

```
2 SELECT
3 cus.customer_state,
4 ROUND(SUM(price),2) AS total_price,
5 ROUND(AVG(PRICE),2) AS avg_price
6 FROM <u>'target.order_item'</u> AS ordit
7 JOIN <u>'target.orders'</u> AS ord
8 ON ordit.order_id = ord.order_id
9 JOIN 'target.customer' AS cus
10 ON ord.customer_id = cus.customer_id
11 GROUP BY cus.customer_state
12 ORDER BY total_price DESC
13
Query results
```

JOB INFORMATION RESULTS		CHART J	SON EXECUTION DETA
Row /	customer_state ▼	total_price ▼	avg_price ▼ //
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27

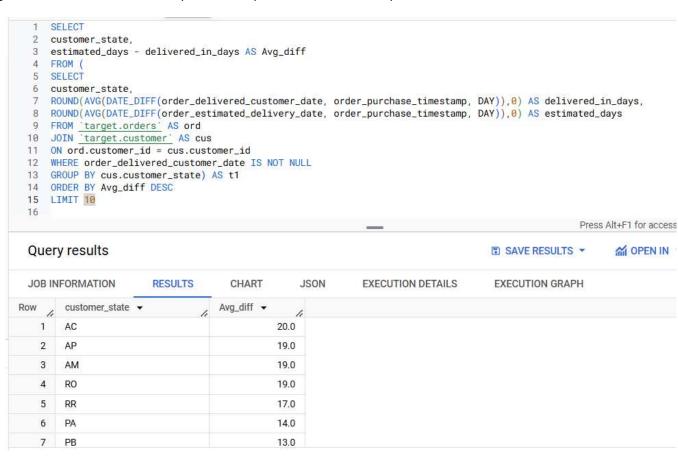
9) Calculating total & average value of freight amount of each state



10) Analysing number of days taken to deliver an Order And finding the Difference in days between Actual and Estimated delivery days

```
SELECT-
  2
  3
       expected_days - days_to_deliver AS diff_in_days
  4
  5
   6
      SELECT
  7
       order_id.
  8
       -DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day) AS-days_to_deliver,
  9
       -DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,day) -AS expected_days
  10
  11
      SELECT *
  12
      FROM 'taxaat asdass'
 Query results
  JOB INFORMATION
                         RESULTS
                                        CHART
                                                     JSON
                                                                 EXECUTION DETAILS
                                                                                          EXECUTION GRAPH
                                      days_to_deliver 🕶
Row /
                                                        expected_days ▼/
       order_id ▼
                                                                          diff_in_days ▼
    1 1b3190b2dfa9d789e1f14c05b...
                                                 208
                                                                    19
                                                                                     -189
    2
       ca07593549f1816d26a572e06...
                                                 209
                                                                    28
                                                                                     -181
    3
      47b40429ed8cce3aee9199792...
                                                 191
                                                                    15
                                                                                     -176
    4
       2fe324febf907e3ea3f2aa9650...
                                                 189
                                                                    22
                                                                                     -167
                                                 194
    5
        285ab9426d6982034523a855f...
                                                                    28
                                                                                     -166
    6
        440d0d17af552815d15a9e41a...
                                                 195
                                                                    30
                                                                                     -165
    7
        c27815f7e3dd0b926b5855262...
                                                 187
                                                                    25
                                                                                     -162
        0f4519c5f1c541ddec9f21b3bd...
    8
                                                 194
                                                                    32
                                                                                     -162
                                                 175
    9
        d24e8541128cea179a11a6517...
                                                                    13
                                                                                     -162
```

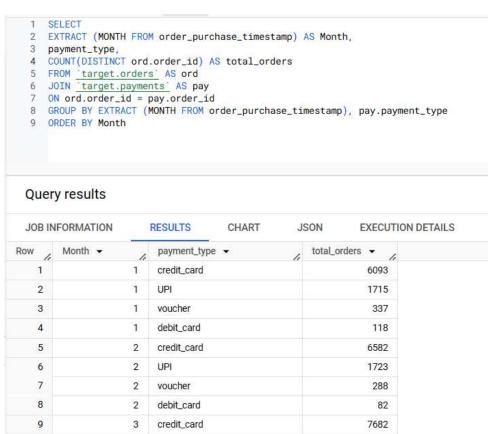
11) Finding TOP 10 states where order delivery is faster compared to estimated delivery date



12) Finding top 5 Countries with highest average freight value

```
1 WITH main AS (
     SELECT *
  3 FROM <u>'target.order_item'</u> AS ordit
  4 JOIN 'target.orders' AS ord
  5 ON ordit.order_id = ord.order_id
  6 JOIN <u>'target.customer'</u> AS cus
  7 ON cus.customer_id = ord.customer_id)
  8
  9 SELECT
 10
      customer_state,
       round(AVG(freight_value),2) AS average_freight
 11
 12 FROM main
 13 GROUP BY customer_state
 14 ORDER BY average_freight DESC
 15 LIMIT 5
 16
 Query results
 JOB INFORMATION
                        RESULTS
                                                  JSON
                                                              EXECUTION DETAILS
                                      CHART
                                    average_freight •
Row /
       customer_state -
        RR
   1
                                             42.98
   2
        PB
                                             42.72
   3
       RO
                                             41.07
   4
        AC
                                             40.07
   5
        PI
                                             39.15
```

13) Finding Number of Order placed Month on month Using different payment types



Insights: From the above output we can See that Consumers prefer Credit Card payment method over the others

14) Finding Total Number of order places on Basis of Payment Installation

