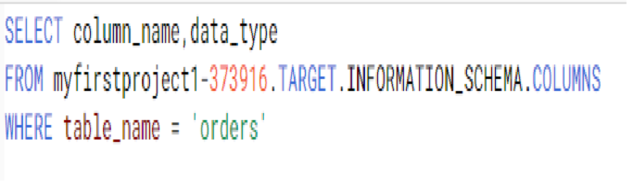
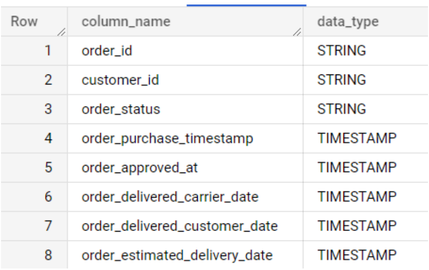
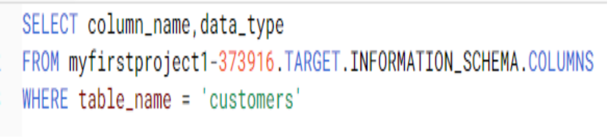
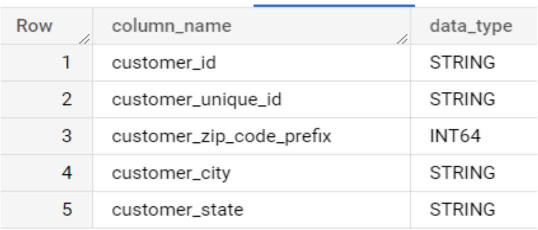
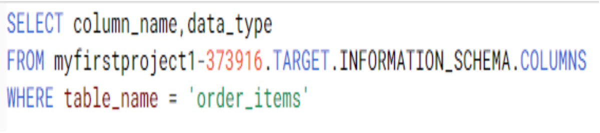
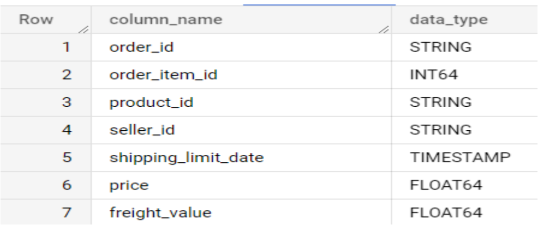
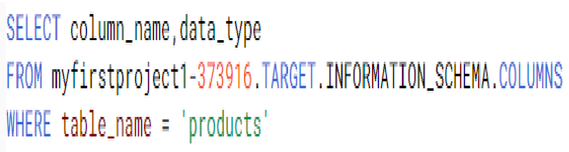
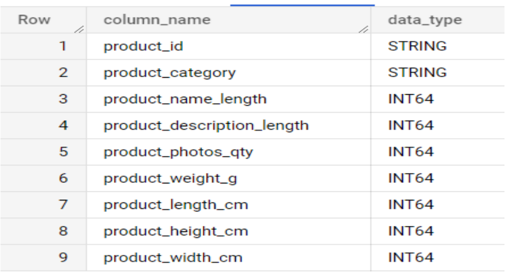
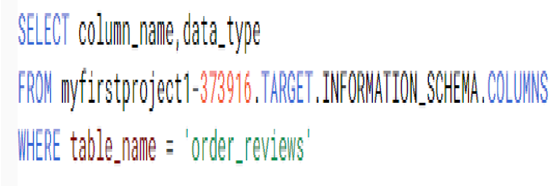
# TARGET

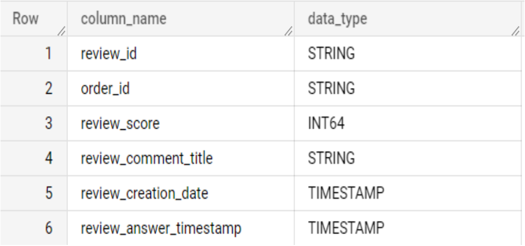
# 1 IMPORTING DATA & EXPLORATORY ANALYSIS

## 1.1 Data types of columns in a table

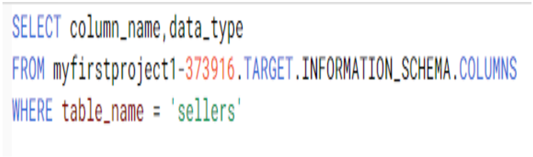
<table>
<tr><th>QUERY</th><th>QUERY_RESULT</th></tr>
<tr><td>

<pre>
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'orders'
</pre>
</td><td>

<table>
<tr><th>Row</th><th>column_name</th><th>data_type</th></tr>
<tr><td>1</td><td>order_id</td><td>STRING</td></tr>
<tr><td>2</td><td>customer_id</td><td>STRING</td></tr>
<tr><td>3</td><td>order_status</td><td>STRING</td></tr>
<tr><td>4</td><td>order_purchase_timestamp</td><td>TIMESTAMP</td></tr>
<tr><td>5</td><td>order_approved_at</td><td>TIMESTAMP</td></tr>
<tr><td>6</td><td>order_delivered_carrier_date</td><td>TIMESTAMP</td></tr>
<tr><td>7</td><td>order_delivered_customer_date</td><td>TIMESTAMP</td></tr>
<tr><td>8</td><td>order_estimated_delivery_date</td><td>TIMESTAMP</td></tr>
</table>
</td></tr>
<tr><td>

<pre>
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
</pre>
</td><td>

<table>
<tr><th>Row</th><th>column_name</th><th>data_type</th></tr>
<tr><td>1</td><td>customer_id</td><td>STRING</td></tr>
<tr><td>2</td><td>customer_unique_id</td><td>STRING</td></tr>
<tr><td>3</td><td>customer_zip_code_prefix</td><td>INT64</td></tr>
<tr><td>4</td><td>customer_city</td><td>STRING</td></tr>
<tr><td>5</td><td>customer_state</td><td>STRING</td></tr>
</table>
</td></tr>
<tr><td>

<pre>
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'order_items'
</pre>
</td><td>

<table>
<tr><th>Row</th><th>column_name</th><th>data_type</th></tr>
<tr><td>1</td><td>order_id</td><td>STRING</td></tr>
<tr><td>2</td><td>order_item_id</td><td>INT64</td></tr>
<tr><td>3</td><td>product_id</td><td>STRING</td></tr>
<tr><td>4</td><td>seller_id</td><td>STRING</td></tr>
<tr><td>5</td><td>shipping_limit_date</td><td>TIMESTAMP</td></tr>
<tr><td>6</td><td>price</td><td>FLOAT64</td></tr>
<tr><td>7</td><td>freight_value</td><td>FLOAT64</td></tr>
</table>
</td></tr>
<tr><td>

<pre>
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'products'
</pre>
</td><td>

<table>
<tr><th>Row</th><th>column_name</th><th>data_type</th></tr>
<tr><td>1</td><td>product_id</td><td>STRING</td></tr>
<tr><td>2</td><td>product_category</td><td>STRING</td></tr>
<tr><td>3</td><td>product_name_length</td><td>INT64</td></tr>
<tr><td>4</td><td>product_description_length</td><td>INT64</td></tr>
<tr><td>5</td><td>product_photos_qty</td><td>INT64</td></tr>
<tr><td>6</td><td>product_weight_g</td><td>INT64</td></tr>
<tr><td>7</td><td>product_length_cm</td><td>INT64</td></tr>
<tr><td>8</td><td>product_height_cm</td><td>INT64</td></tr>
<tr><td>9</td><td>product_width_cm</td><td>INT64</td></tr>
</table>
</td></tr>
</table>



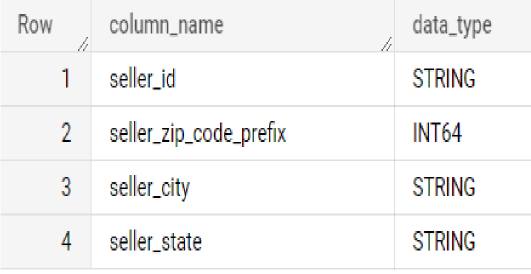
```sql
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'order_reviews'
```



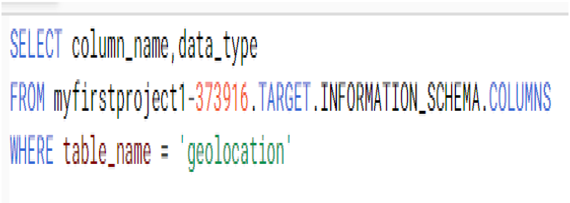
| Row | column_name | data_type |
|---|---|---|
| 1 | review_id | STRING |
| 2 | order_id | STRING |
| 3 | review_score | INT64 |
| 4 | review_comment_title | STRING |
| 5 | review_creation_date | TIMESTAMP |
| 6 | review_answer_timestamp | TIMESTAMP |



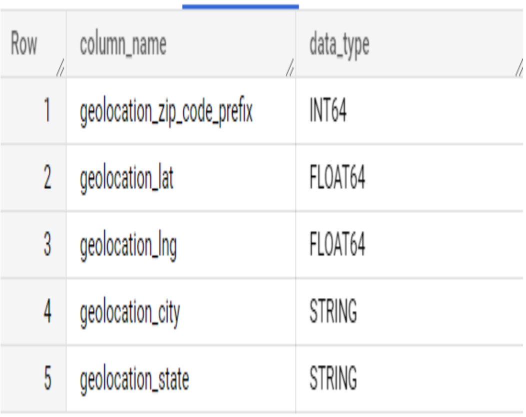
```sql
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'sellers'
```



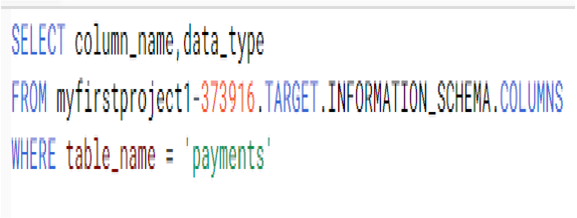
| Row | column_name | data_type |
|---|---|---|
| 1 | seller_id | STRING |
| 2 | seller_zip_code_prefix | INT64 |
| 3 | seller_city | STRING |
| 4 | seller_state | STRING |



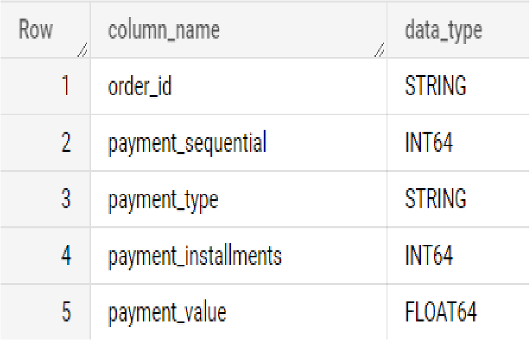
```sql
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'geolocation'
```



| Row | column_name | data_type |
|---|---|---|
| 1 | geolocation_zip_code_prefix | INT64 |
| 2 | geolocation_lat | FLOAT64 |
| 3 | geolocation_lng | FLOAT64 |
| 4 | geolocation_city | STRING |
| 5 | geolocation_state | STRING |

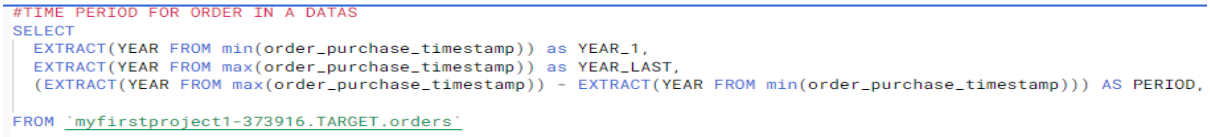


```sql
SELECT column_name,data_type
FROM myfirstproject1-373916.TARGET.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'payments'
```



| Row | column_name | data_type |
|---|---|---|
| 1 | order_id | STRING |
| 2 | payment_sequential | INT64 |
| 3 | payment_type | STRING |
| 4 | payment_installments | INT64 |
| 5 | payment_value | FLOAT64 |

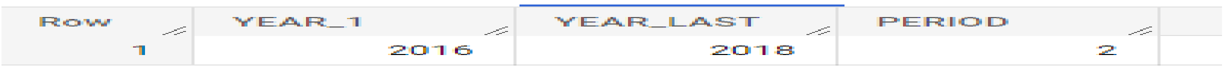
## 1.2 Time period for which the data is given



```
#TIME PERIOD FOR ORDER IN A DATAS
SELECT
  EXTRACT(YEAR FROM min(order_purchase_timestamp)) as YEAR_1,
  EXTRACT(YEAR FROM max(order_purchase_timestamp)) as YEAR_LAST,
  (EXTRACT(YEAR FROM max(order_purchase_timestamp)) - EXTRACT(YEAR FROM min(order_purchase_timestamp))) AS PERIOD,

FROM `myfirstproject1-373916.TARGET.orders`
```

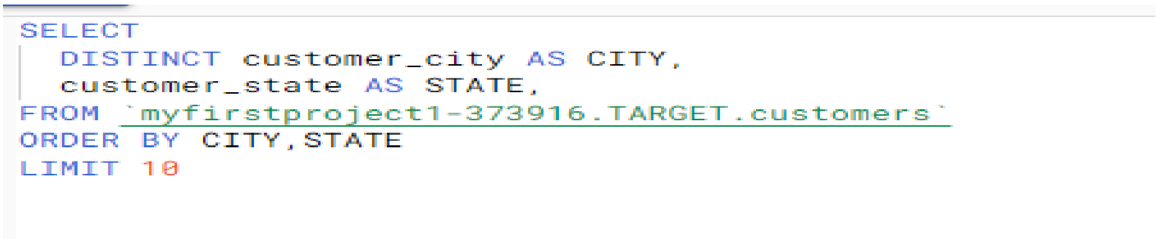
*Fig:* ***QUERY***



| Row | YEAR_1 | YEAR_LAST | PERIOD |
|---|---|---|---|
| 1 | 2016 | 2018 | 2 |

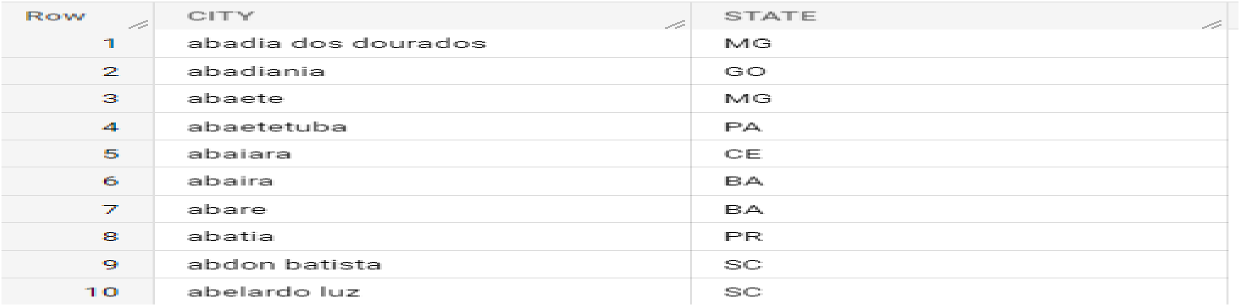
*Fig:* ***QUERY_RESULT***

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



```
SELECT
  DISTINCT customer_city AS CITY,
  customer_state AS STATE,
FROM `myfirstproject1-373916.TARGET.customers`
ORDER BY CITY,STATE
LIMIT 10
```

*Fig:* ***QUERY***

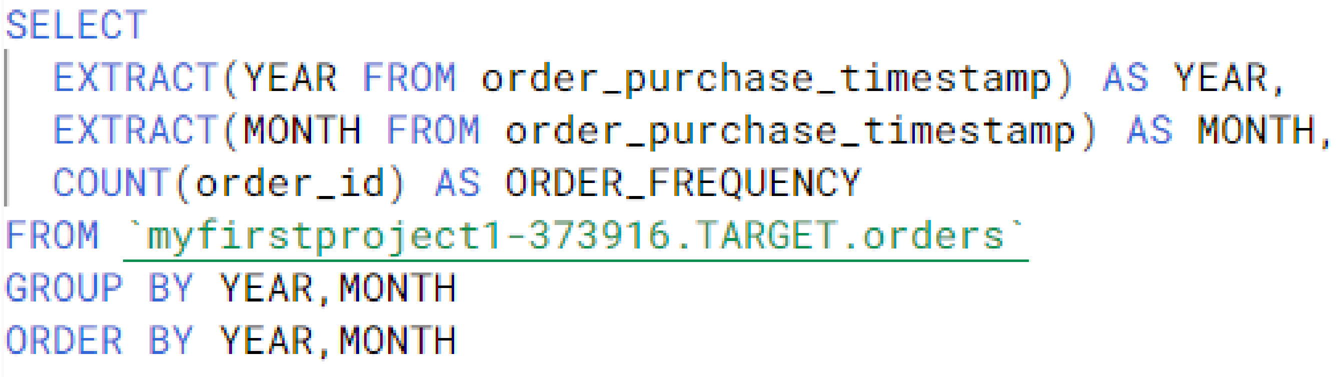


| Row | CITY | STATE |
|---|---|---|
| 1 | abadia dos dourados | MG |
| 2 | abadiania | GO |
| 3 | abaete | MG |
| 4 | abaetetuba | PA |
| 5 | abaiara | CE |
| 6 | abaira | BA |
| 7 | abare | BA |
| 8 | abatia | PR |
| 9 | abdon batista | SC |
| 10 | abelardo luz | SC |

*Fig:* ***QUERY_RESULT***

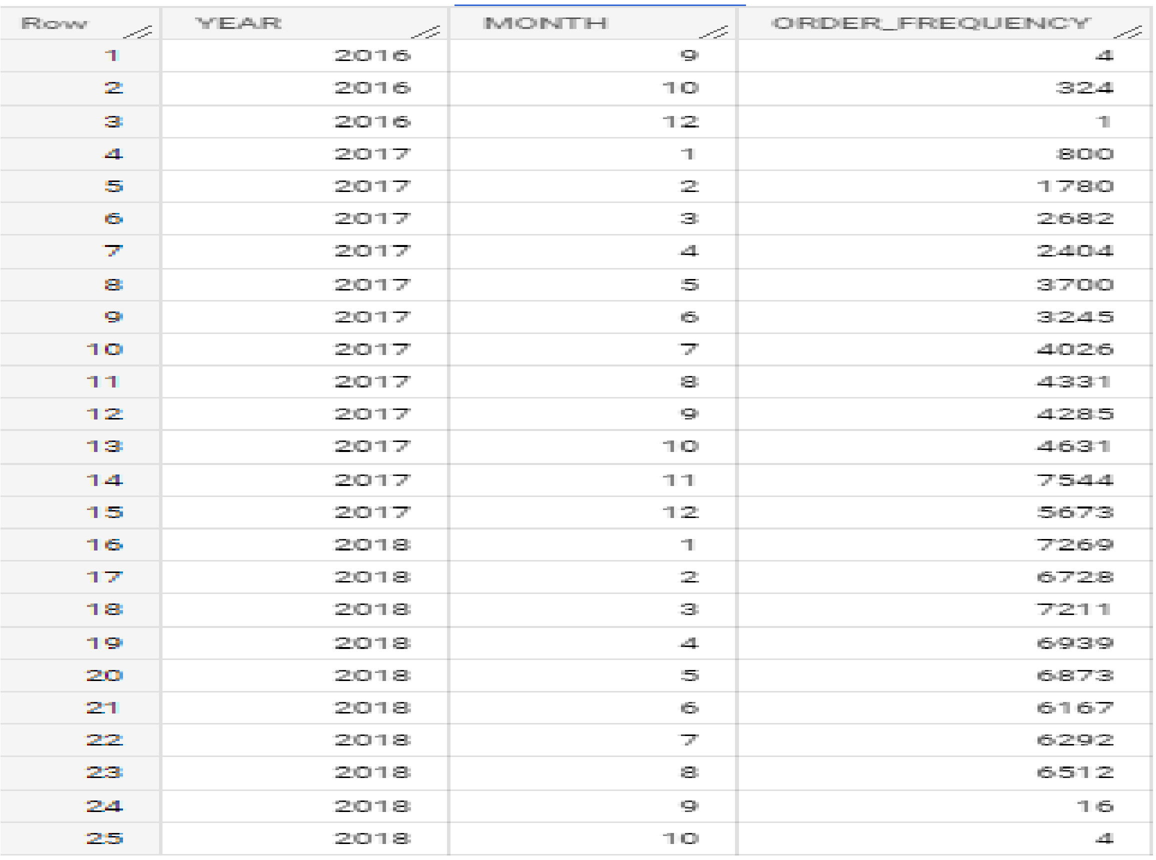
# 2 IN-DEPTH EXPLORATION

## 2.1 Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks as specific months?



```
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS YEAR,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS MONTH,
  COUNT(order_id) AS ORDER_FREQUENCY
FROM `myfirstproject1-373916.TARGET.orders`
GROUP BY YEAR,MONTH
ORDER BY YEAR,MONTH
```

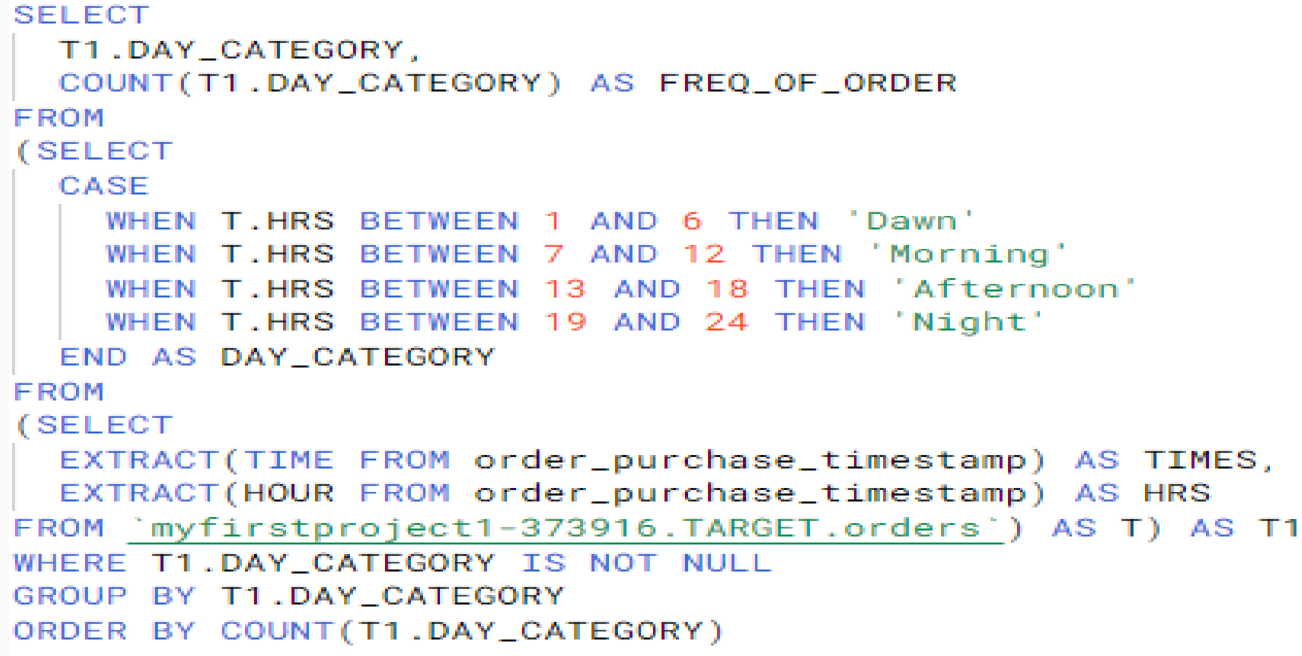
*Fig:* ***QUERY***



| Row | YEAR | MONTH | ORDER_FREQUENCY |
|---|---|---|---|
| 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 |
| 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 |

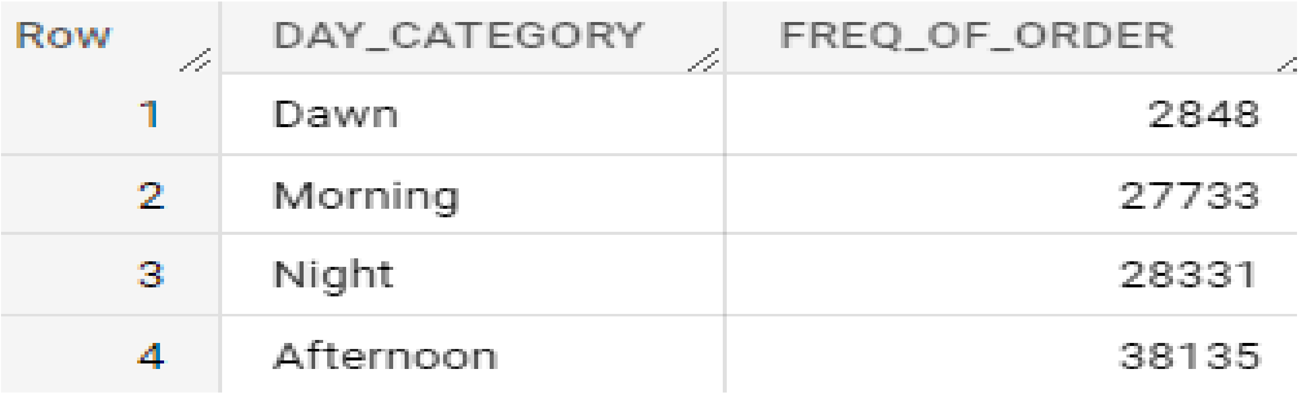
*Fig: **QUERY_RESULT & FIG***

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



```
SELECT
  T1.DAY_CATEGORY,
  COUNT(T1.DAY_CATEGORY) AS FREQ_OF_ORDER
FROM
(SELECT
  CASE
    WHEN T.HRS BETWEEN 1 AND 6 THEN 'Dawn'
    WHEN T.HRS BETWEEN 7 AND 12 THEN 'Morning'
    WHEN T.HRS BETWEEN 13 AND 18 THEN 'Afternoon'
    WHEN T.HRS BETWEEN 19 AND 24 THEN 'Night'
  END AS DAY_CATEGORY
FROM
(SELECT
  EXTRACT(TIME FROM order_purchase_timestamp) AS TIMES,
  EXTRACT(HOUR FROM order_purchase_timestamp) AS HRS
FROM `myfirstproject1-373916.TARGET.orders`) AS T) AS T1
WHERE T1.DAY_CATEGORY IS NOT NULL
GROUP BY T1.DAY_CATEGORY
ORDER BY COUNT(T1.DAY_CATEGORY)
```

*Fig: **QUERY***

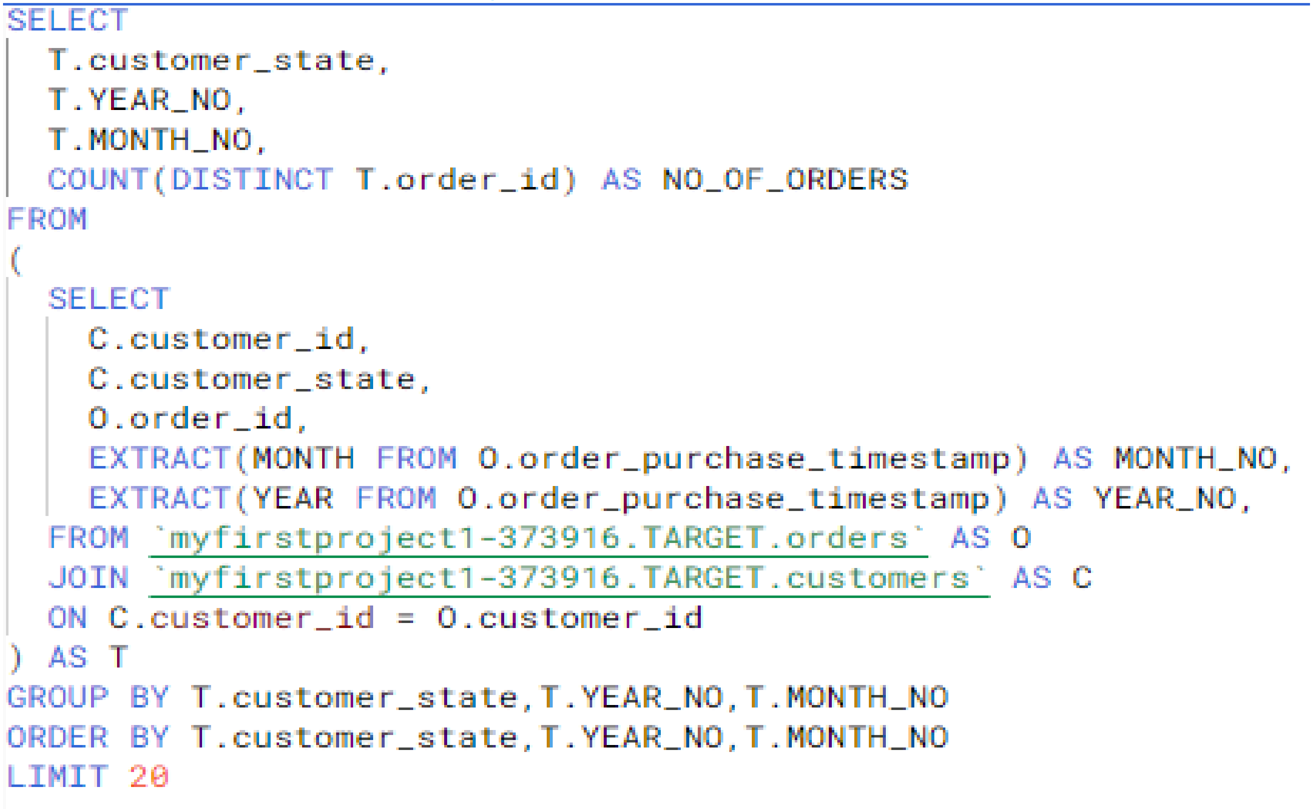


| Row | DAY_CATEGORY | FREQ_OF_ORDER |
|---|---|---|
| 1 | Dawn | 2848 |
| 2 | Morning | 27733 |
| 3 | Night | 28331 |
| 4 | Afternoon | 38135 |

Fig: ***QUERY_RESULT & FIG***

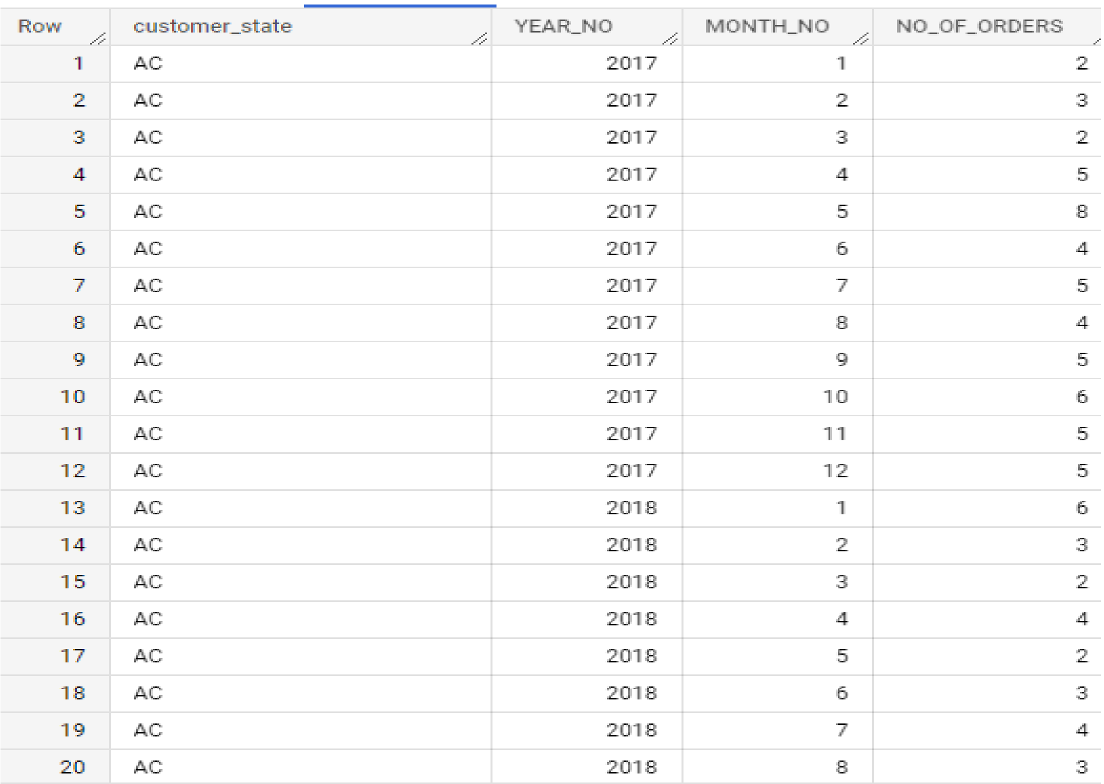
# 3 EVOLUTION OF E-COMMERCE ORDERS IN THE BRAZIL REGION:

## 3.1 Get month on month orders by states:



```
SELECT
  T.customer_state,
  T.YEAR_NO,
  T.MONTH_NO,
  COUNT(DISTINCT T.order_id) AS NO_OF_ORDERS
FROM
(
  SELECT
    C.customer_id,
    C.customer_state,
    O.order_id,
    EXTRACT(MONTH FROM O.order_purchase_timestamp) AS MONTH_NO,
    EXTRACT(YEAR FROM O.order_purchase_timestamp) AS YEAR_NO,
  FROM `myfirstproject1-373916.TARGET.orders` AS O
  JOIN `myfirstproject1-373916.TARGET.customers` AS C
  ON C.customer_id = O.customer_id
) AS T
GROUP BY T.customer_state,T.YEAR_NO,T.MONTH_NO
ORDER BY T.customer_state,T.YEAR_NO,T.MONTH_NO
LIMIT 20
```

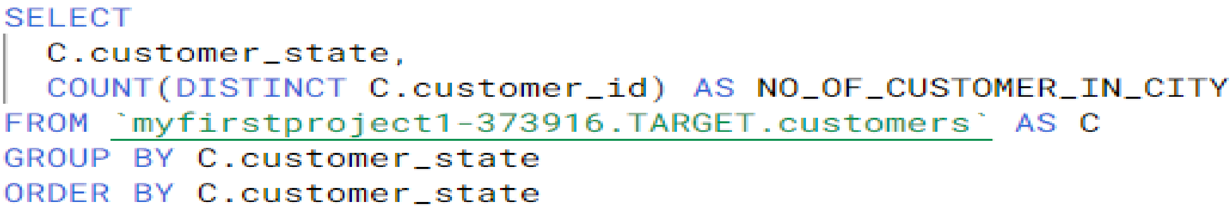
Fig: ***QUERY***



| Row | customer_state | YEAR_NO | MONTH_NO | NO_OF_ORDERS |
|---|---|---|---|---|
| 1 | AC | 2017 | 1 | 2 |
| 2 | AC | 2017 | 2 | 3 |
| 3 | AC | 2017 | 3 | 2 |
| 4 | AC | 2017 | 4 | 5 |
| 5 | AC | 2017 | 5 | 8 |
| 6 | AC | 2017 | 6 | 4 |
| 7 | AC | 2017 | 7 | 5 |
| 8 | AC | 2017 | 8 | 4 |
| 9 | AC | 2017 | 9 | 5 |
| 10 | AC | 2017 | 10 | 6 |
| 11 | AC | 2017 | 11 | 5 |
| 12 | AC | 2017 | 12 | 5 |
| 13 | AC | 2018 | 1 | 6 |
| 14 | AC | 2018 | 2 | 3 |
| 15 | AC | 2018 | 3 | 2 |
| 16 | AC | 2018 | 4 | 4 |
| 17 | AC | 2018 | 5 | 2 |
| 18 | AC | 2018 | 6 | 3 |
| 19 | AC | 2018 | 7 | 4 |
| 20 | AC | 2018 | 8 | 3 |

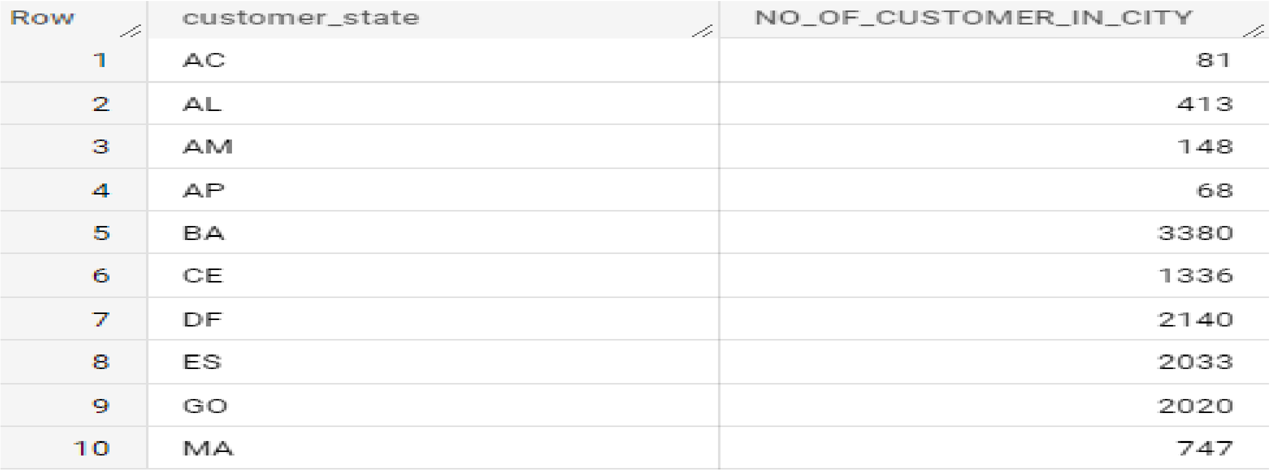
*Fig*: ***QUERY_RESULT***

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



```
SELECT
  C.customer_state,
  COUNT(DISTINCT C.customer_id) AS NO_OF_CUSTOMER_IN_CITY
FROM `myfirstproject1-373916.TARGET.customers` AS C
GROUP BY C.customer_state
ORDER BY C.customer_state
```

*Fig: **QUERY***

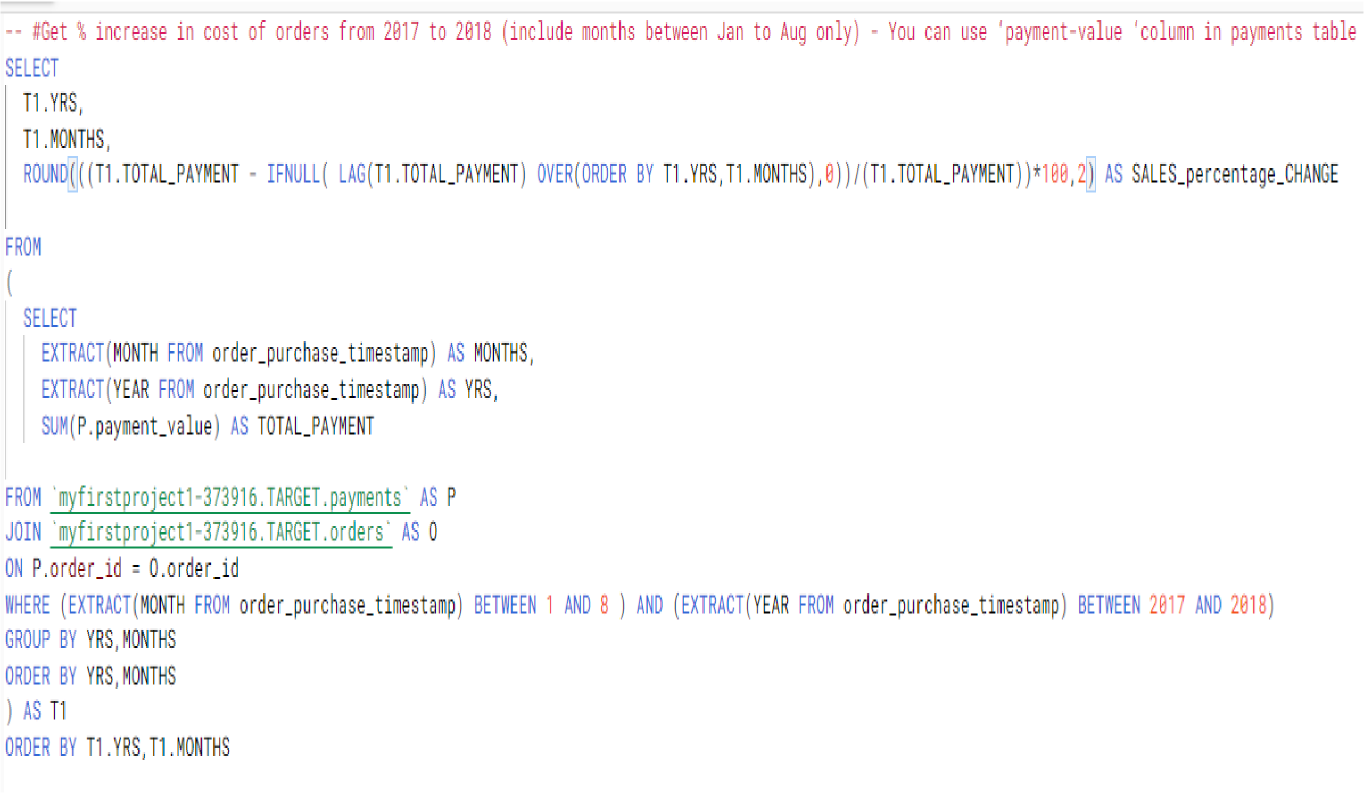


| Row | customer_state | NO_OF_CUSTOMER_IN_CITY |
|---|---|---|
| 1 | AC | 81 |
| 2 | AL | 413 |
| 3 | AM | 148 |
| 4 | AP | 68 |
| 5 | BA | 3380 |
| 6 | CE | 1336 |
| 7 | DF | 2140 |
| 8 | ES | 2033 |
| 9 | GO | 2020 |
| 10 | MA | 747 |

*Fig*: ***QUERY_RESULT***

# 4 IMPACT ON ECONOMY:

## 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

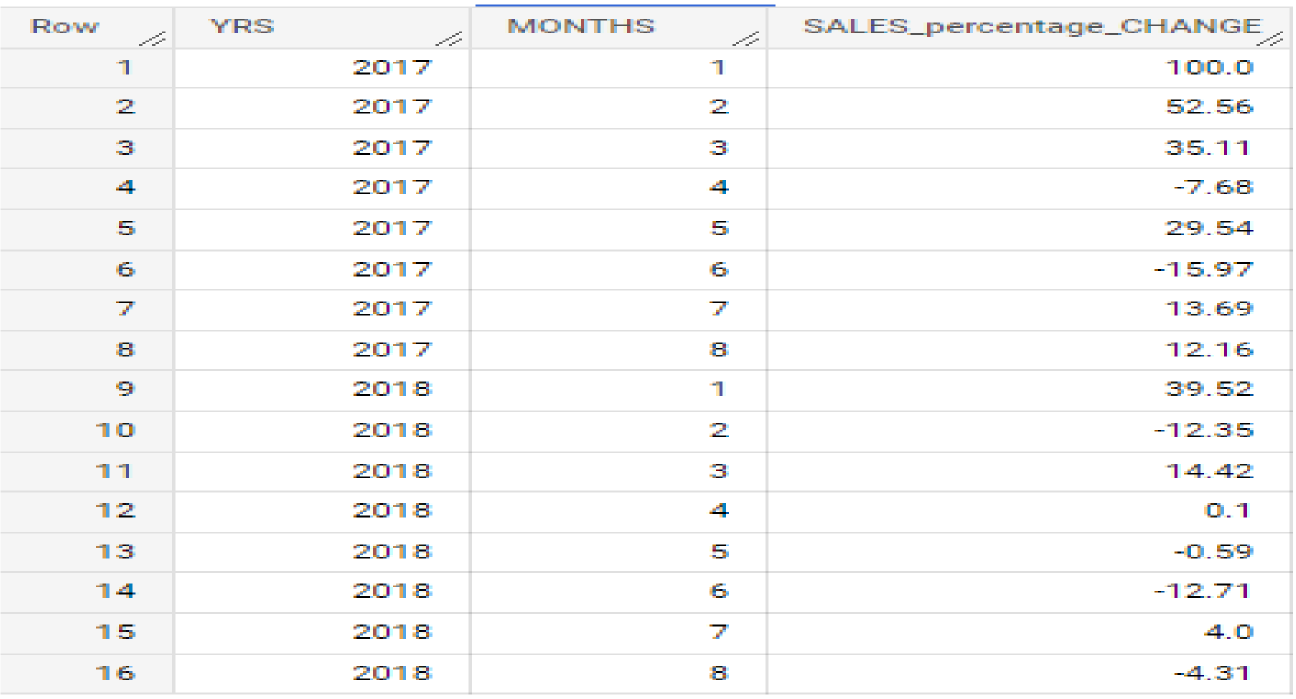


```
-- #Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use 'payment-value 'column in payments table
SELECT
  T1.YRS,
  T1.MONTHS,
  ROUND(((T1.TOTAL_PAYMENT - IFNULL( LAG(T1.TOTAL_PAYMENT) OVER(ORDER BY T1.YRS,T1.MONTHS),0))/(T1.TOTAL_PAYMENT))*100,2) AS SALES_percentage_CHANGE

FROM
(
  SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) AS MONTHS,
    EXTRACT(YEAR FROM order_purchase_timestamp) AS YRS,
    SUM(P.payment_value) AS TOTAL_PAYMENT

FROM `myfirstproject1-373916.TARGET.payments` AS P
JOIN `myfirstproject1-373916.TARGET.orders` AS O
ON P.order_id = O.order_id
WHERE (EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8 ) AND (EXTRACT(YEAR FROM order_purchase_timestamp) BETWEEN 2017 AND 2018)
GROUP BY YRS,MONTHS
ORDER BY YRS,MONTHS
) AS T1
ORDER BY T1.YRS,T1.MONTHS
```

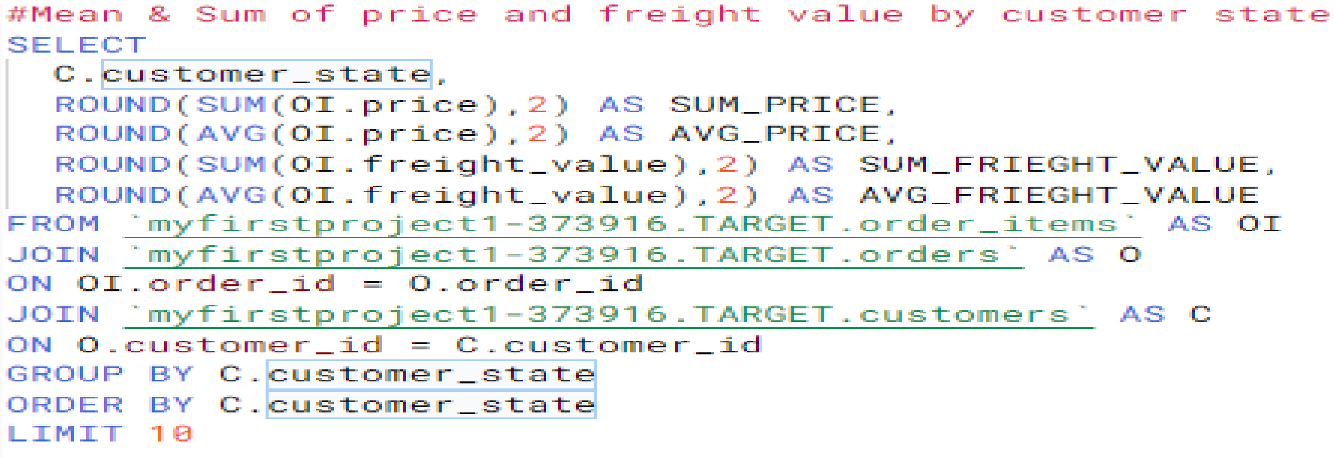
*Fig: **QUERY***



| Row | YRS | MONTHS | SALES_percentage_CHANGE |
|---|---|---|---|
| 1 | 2017 | 1 | 100.0 |
| 2 | 2017 | 2 | 52.56 |
| 3 | 2017 | 3 | 35.11 |
| 4 | 2017 | 4 | -7.68 |
| 5 | 2017 | 5 | 29.54 |
| 6 | 2017 | 6 | -15.97 |
| 7 | 2017 | 7 | 13.69 |
| 8 | 2017 | 8 | 12.16 |
| 9 | 2018 | 1 | 39.52 |
| 10 | 2018 | 2 | -12.35 |
| 11 | 2018 | 3 | 14.42 |
| 12 | 2018 | 4 | 0.1 |
| 13 | 2018 | 5 | -0.59 |
| 14 | 2018 | 6 | -12.71 |
| 15 | 2018 | 7 | 4.0 |
| 16 | 2018 | 8 | -4.31 |

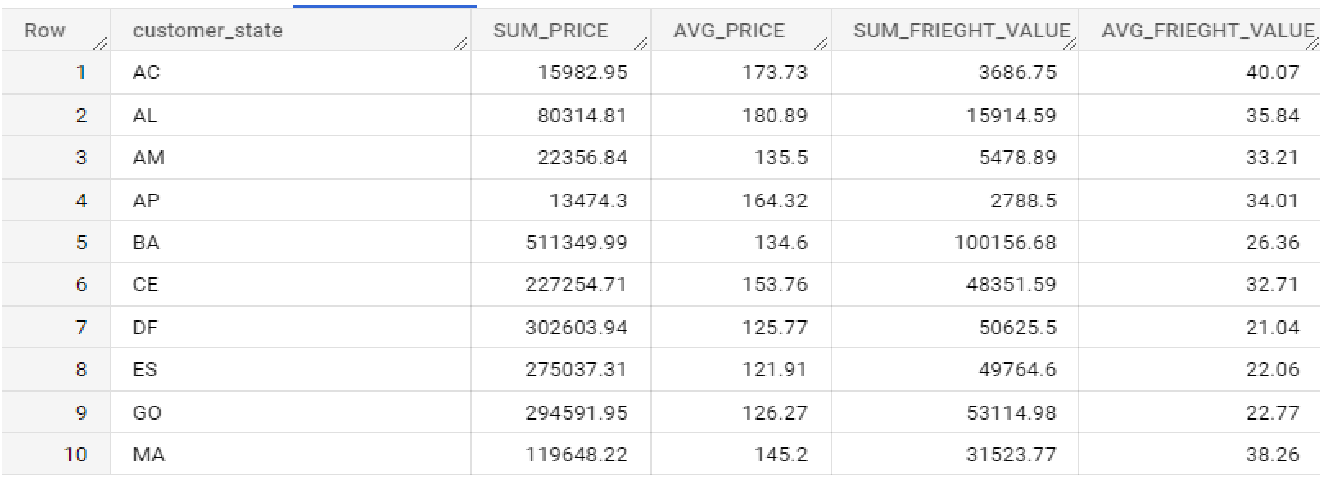
*Fig*: ***QUERY_RESULT***

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



```
#Mean & Sum of price and freight value by customer state
SELECT
  C.customer_state,
  ROUND(SUM(OI.price),2) AS SUM_PRICE,
  ROUND(AVG(OI.price),2) AS AVG_PRICE,
  ROUND(SUM(OI.freight_value),2) AS SUM_FRIEGHT_VALUE,
  ROUND(AVG(OI.freight_value),2) AS AVG_FRIEGHT_VALUE
FROM `myfirstproject1-373916.TARGET.order_items` AS OI
JOIN `myfirstproject1-373916.TARGET.orders` AS O
ON OI.order_id = O.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY C.customer_state
LIMIT 10
```

*Fig: **QUERY***

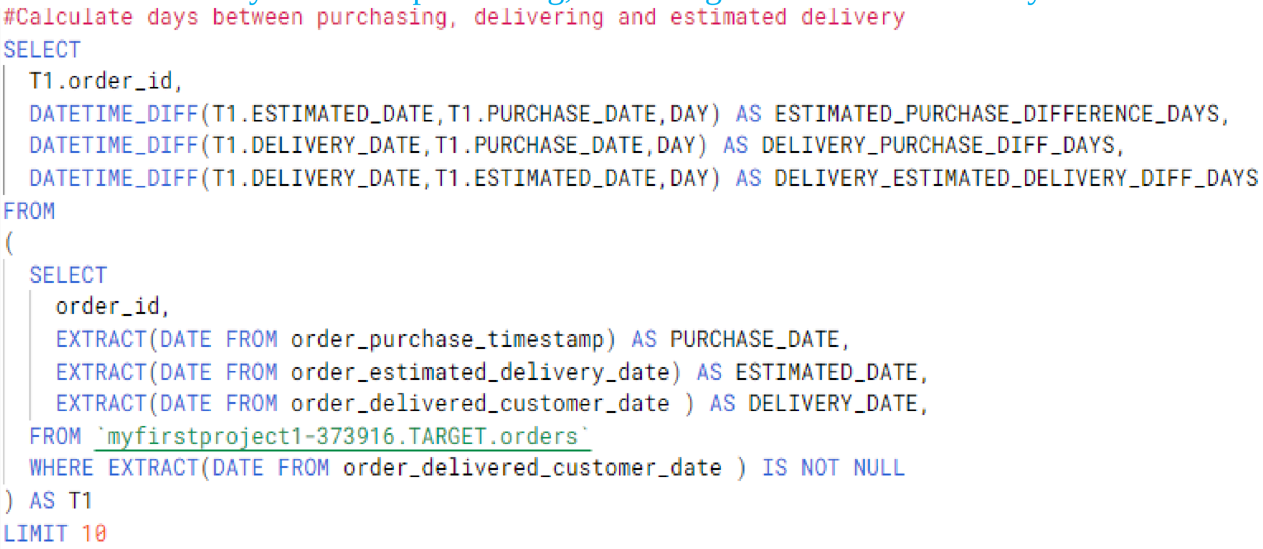


| Row | customer_state | SUM_PRICE | AVG_PRICE | SUM_FRIEGHT_VALUE | AVG_FRIEGHT_VALUE |
|---|---|---|---|---|---|
| 1 | AC | 15982.95 | 173.73 | 3686.75 | 40.07 |
| 2 | AL | 80314.81 | 180.89 | 15914.59 | 35.84 |
| 3 | AM | 22356.84 | 135.5 | 5478.89 | 33.21 |
| 4 | AP | 13474.3 | 164.32 | 2788.5 | 34.01 |
| 5 | BA | 511349.99 | 134.6 | 100156.68 | 26.36 |
| 6 | CE | 227254.71 | 153.76 | 48351.59 | 32.71 |
| 7 | DF | 302603.94 | 125.77 | 50625.5 | 21.04 |
| 8 | ES | 275037.31 | 121.91 | 49764.6 | 22.06 |
| 9 | GO | 294591.95 | 126.27 | 53114.98 | 22.77 |
| 10 | MA | 119648.22 | 145.2 | 31523.77 | 38.26 |

*Fig*: ***QUERY_RESULT***

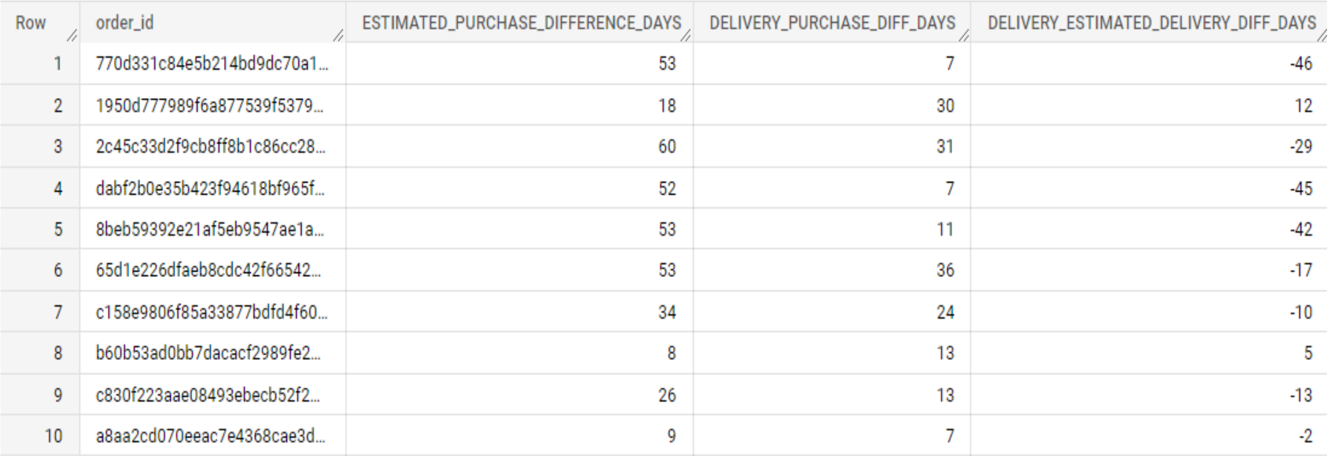
# 5 ANALYSIS ON SALES, FREIGHT AND DELIVERY TIME:

## 5.1 Calculate days between purchasing, delivering and estimated delivery



```
#Calculate days between purchasing, delivering and estimated delivery
SELECT
  T1.order_id,
  DATETIME_DIFF(T1.ESTIMATED_DATE,T1.PURCHASE_DATE,DAY) AS ESTIMATED_PURCHASE_DIFFERENCE_DAYS,
  DATETIME_DIFF(T1.DELIVERY_DATE,T1.PURCHASE_DATE,DAY) AS DELIVERY_PURCHASE_DIFF_DAYS,
  DATETIME_DIFF(T1.DELIVERY_DATE,T1.ESTIMATED_DATE,DAY) AS DELIVERY_ESTIMATED_DELIVERY_DIFF_DAYS
FROM
(
  SELECT
    order_id,
    EXTRACT(DATE FROM order_purchase_timestamp) AS PURCHASE_DATE,
    EXTRACT(DATE FROM order_estimated_delivery_date) AS ESTIMATED_DATE,
    EXTRACT(DATE FROM order_delivered_customer_date ) AS DELIVERY_DATE,
  FROM `myfirstproject1-373916.TARGET.orders`
  WHERE EXTRACT(DATE FROM order_delivered_customer_date ) IS NOT NULL
) AS T1
LIMIT 10
```

*Fig: **QUERY***

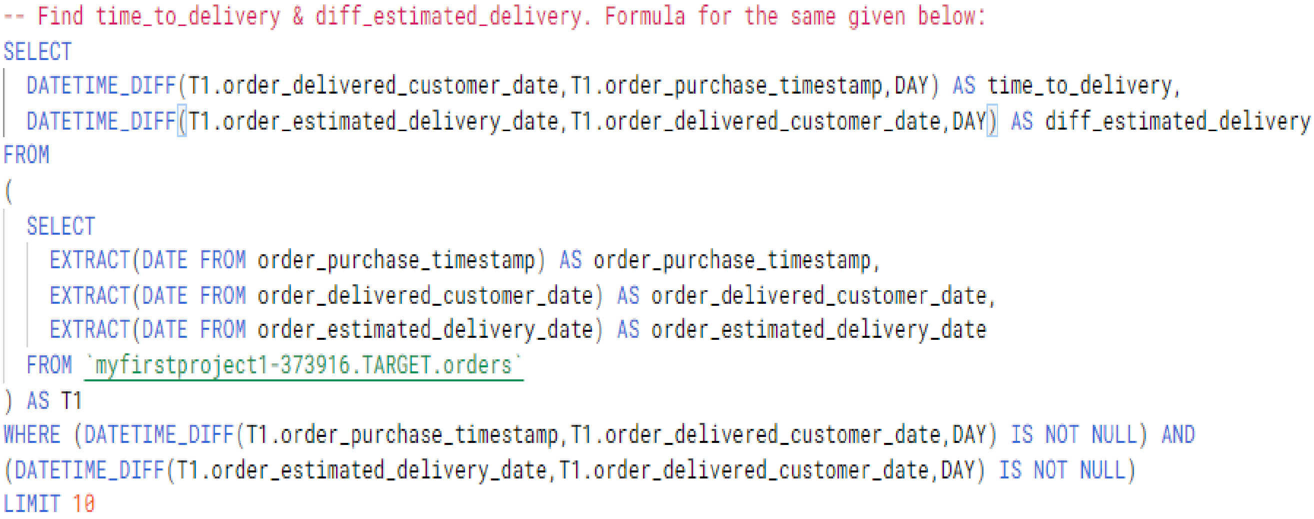


| Row | order_id | ESTIMATED_PURCHASE_DIFFERENCE_DAYS | DELIVERY_PURCHASE_DIFF_DAYS | DELIVERY_ESTIMATED_DELIVERY_DIFF_DAYS |
|---|---|---|---|---|
| 1 | 770d331c84e5b214bd9dc70a1... | 53 | 7 | -46 |
| 2 | 1950d777989f6a877539f5379... | 18 | 30 | 12 |
| 3 | 2c45c33d2f9cb8ff8b1c86cc28... | 60 | 31 | -29 |
| 4 | dabf2b0e35b423f94618bf965f... | 52 | 7 | -45 |
| 5 | 8beb59392e21af5eb9547ae1a... | 53 | 11 | -42 |
| 6 | 65d1e226dfaeb8cdc42f66542... | 53 | 36 | -17 |
| 7 | c158e9806f85a33877bdfd4f60... | 34 | 24 | -10 |
| 8 | b60b53ad0bb7dacacf2989fe2... | 8 | 13 | 5 |
| 9 | c830f223aae08493ebecb52f2... | 26 | 13 | -13 |
| 10 | a8aa2cd070eeac7e4368cae3d... | 9 | 7 | -2 |

*Fig*: ***QUERY_RESULT***

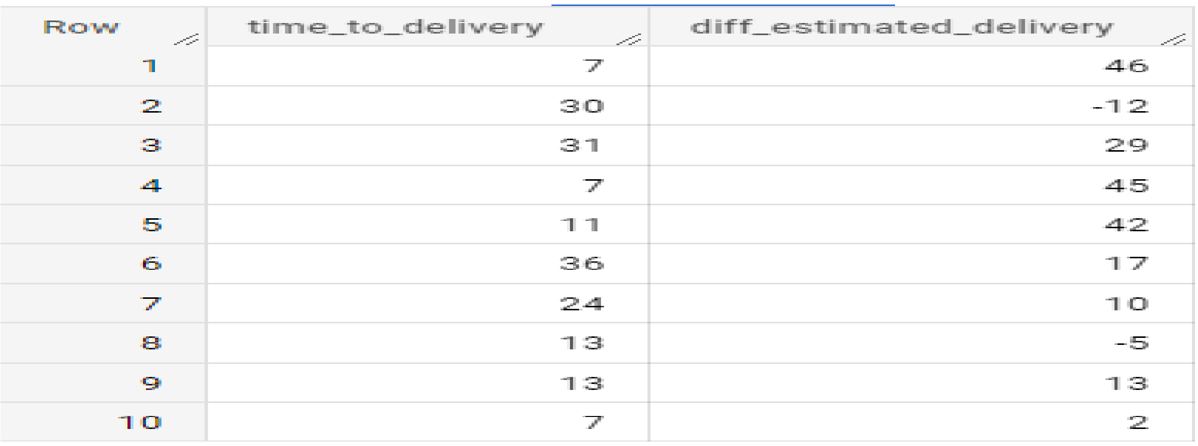
**OUTCOMES:** *Here '- ve' value indicate delivery after the expected estimated date & '+ve' indicate before expected date. All the above values are in days.*

## 5.2 Find time-to-delivery & diff-estimated-delivery. Formula for the same given below:



```
-- Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
SELECT
  DATETIME_DIFF(T1.order_delivered_customer_date,T1.order_purchase_timestamp,DAY) AS time_to_delivery,
  DATETIME_DIFF(T1.order_estimated_delivery_date,T1.order_delivered_customer_date,DAY) AS diff_estimated_delivery
FROM
(
  SELECT
    EXTRACT(DATE FROM order_purchase_timestamp) AS order_purchase_timestamp,
    EXTRACT(DATE FROM order_delivered_customer_date) AS order_delivered_customer_date,
    EXTRACT(DATE FROM order_estimated_delivery_date) AS order_estimated_delivery_date
  FROM `myfirstproject1-373916.TARGET.orders`
) AS T1
WHERE (DATETIME_DIFF(T1.order_purchase_timestamp,T1.order_delivered_customer_date,DAY) IS NOT NULL) AND
(DATETIME_DIFF(T1.order_estimated_delivery_date,T1.order_delivered_customer_date,DAY) IS NOT NULL)
LIMIT 10
```

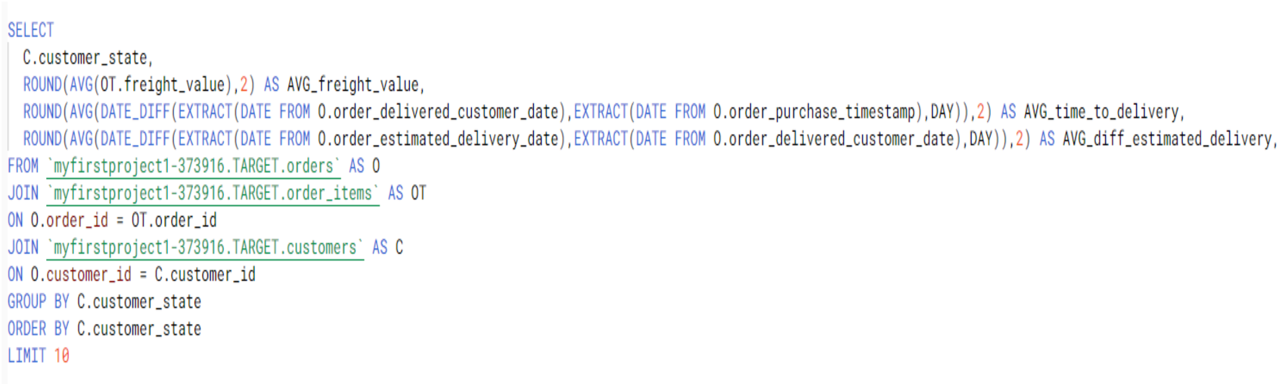
*Fig: **QUERY***



| Row | time_to_delivery | diff_estimated_delivery |
|---|---|---|
| 1 | 7 | 46 |
| 2 | 30 | -12 |
| 3 | 31 | 29 |
| 4 | 7 | 45 |
| 5 | 11 | 42 |
| 6 | 36 | 17 |
| 7 | 24 | 10 |
| 8 | 13 | -5 |
| 9 | 13 | 13 |
| 10 | 7 | 2 |

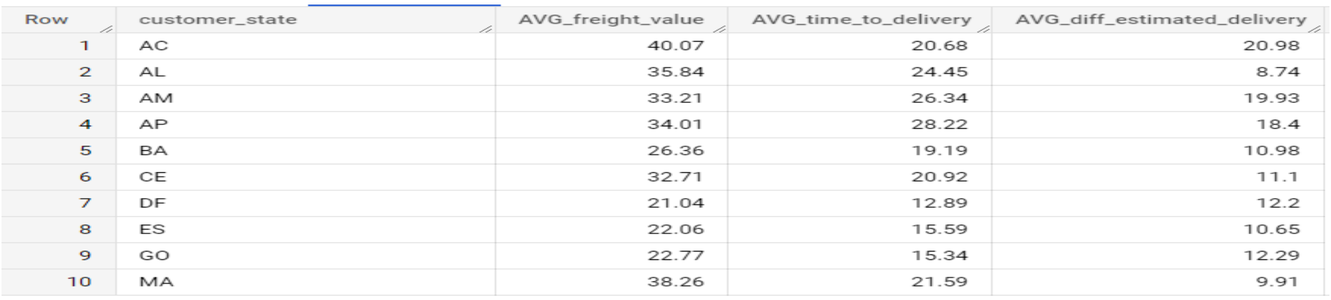
*Fig*: ***QUERY_RESULT***

## 5.3 Group data by state, take mean of freight-value, time-to-delivery, diff-estimate-delivery



```
SELECT
  C.customer_state,
  ROUND(AVG(OT.freight_value),2) AS AVG_freight_value,
  ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM O.order_delivered_customer_date),EXTRACT(DATE FROM O.order_purchase_timestamp),DAY)),2) AS AVG_time_to_delivery,
  ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM O.order_estimated_delivery_date),EXTRACT(DATE FROM O.order_delivered_customer_date),DAY)),2) AS AVG_diff_estimated_delivery,
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.order_items` AS OT
ON O.order_id = OT.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY C.customer_state
LIMIT 10
```

*Fig: **QUERY***

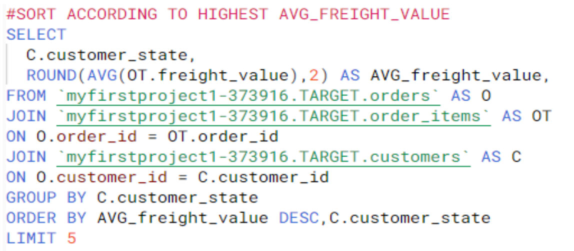


| Row | customer_state | AVG_freight_value | AVG_time_to_delivery | AVG_diff_estimated_delivery |
|---|---|---|---|---|
| 1 | AC | 40.07 | 20.68 | 20.98 |
| 2 | AL | 35.84 | 24.45 | 8.74 |
| 3 | AM | 33.21 | 26.34 | 19.93 |
| 4 | AP | 34.01 | 28.22 | 18.4 |
| 5 | BA | 26.36 | 19.19 | 10.98 |
| 6 | CE | 32.71 | 20.92 | 11.1 |
| 7 | DF | 21.04 | 12.89 | 12.2 |
| 8 | ES | 22.06 | 15.59 | 10.65 |
| 9 | GO | 22.77 | 15.34 | 12.29 |
| 10 | MA | 38.26 | 21.59 | 9.91 |

*Fig*: ***QUERY_RESULT***

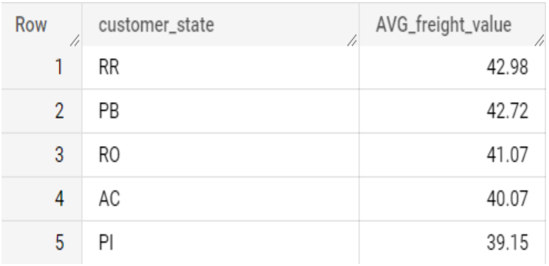
## 5.4 Sort the data to get the following:

## 5.5 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5



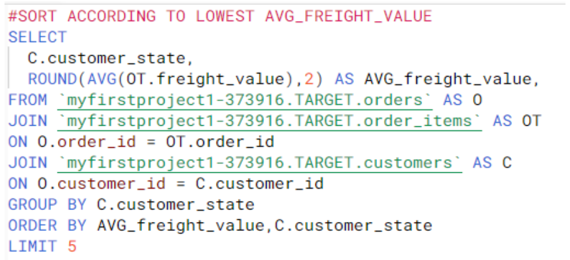
```
#SORT ACCORDING TO HIGHEST AVG_FREIGHT_VALUE
SELECT
  C.customer_state,
  ROUND(AVG(OT.freight_value),2) AS AVG_freight_value,
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.order_items` AS OT
ON O.order_id = OT.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY AVG_freight_value DESC,C.customer_state
LIMIT 5
```

*Fig: **QUERY***



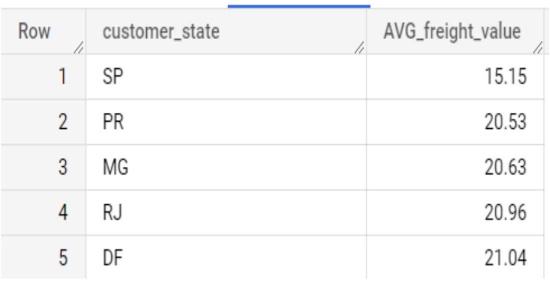
| Row | customer_state | AVG_freight_value |
|---|---|---|
| 1 | RR | 42.98 |
| 2 | PB | 42.72 |
| 3 | RO | 41.07 |
| 4 | AC | 40.07 |
| 5 | PI | 39.15 |

*Fig*: ***QUERY_RESULT***



```
#SORT ACCORDING TO LOWEST AVG_FREIGHT_VALUE
SELECT
  C.customer_state,
  ROUND(AVG(OT.freight_value),2) AS AVG_freight_value,
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.order_items` AS OT
ON O.order_id = OT.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY AVG_freight_value,C.customer_state
LIMIT 5
```

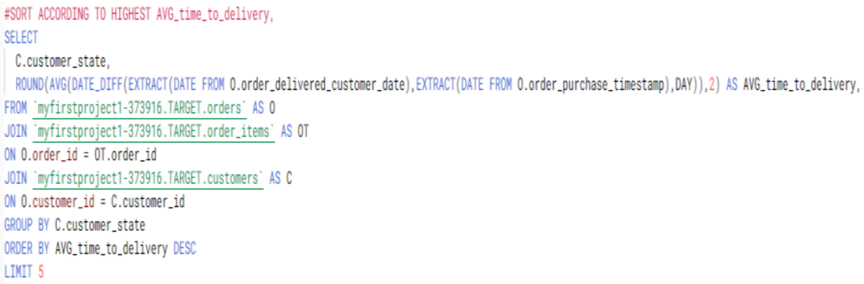
*Fig: **QUERY***



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

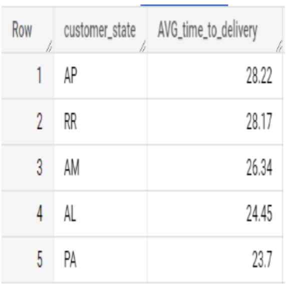
*Fig*: ***QUERY_RESULT***

## 5.6 Top 5 states with highest/lowest average time to delivery



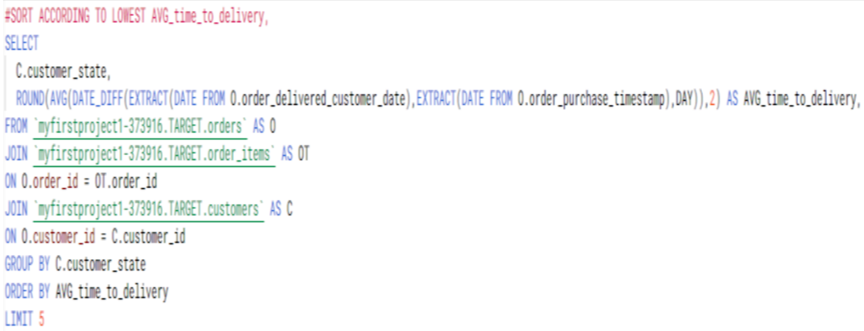
```
#SORT ACCORDING TO HIGHEST AVG_time_to_delivery,
SELECT
  C.customer_state,
  ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM O.order_delivered_customer_date),EXTRACT(DATE FROM O.order_purchase_timestamp),DAY)),2) AS AVG_time_to_delivery,
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.order_items` AS OT
ON O.order_id = OT.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY AVG_time_to_delivery DESC
LIMIT 5
```

*Fig:* ***QUERY***



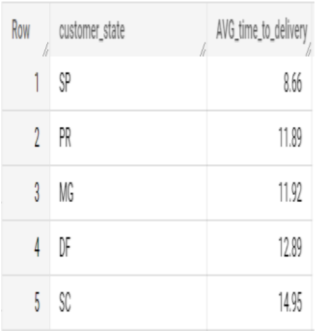
| Row | customer_state | AVG_time_to_delivery |
|---|---|---|
| 1 | AP | 28.22 |
| 2 | RR | 28.17 |
| 3 | AM | 26.34 |
| 4 | AL | 24.45 |
| 5 | PA | 23.7 |

*Fig*: ***QUERY_RESULT***



```
#SORT ACCORDING TO LOWEST AVG_time_to_delivery,
SELECT
  C.customer_state,
  ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM O.order_delivered_customer_date),EXTRACT(DATE FROM O.order_purchase_timestamp),DAY)),2) AS AVG_time_to_delivery,
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.order_items` AS OT
ON O.order_id = OT.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY AVG_time_to_delivery
LIMIT 5
```

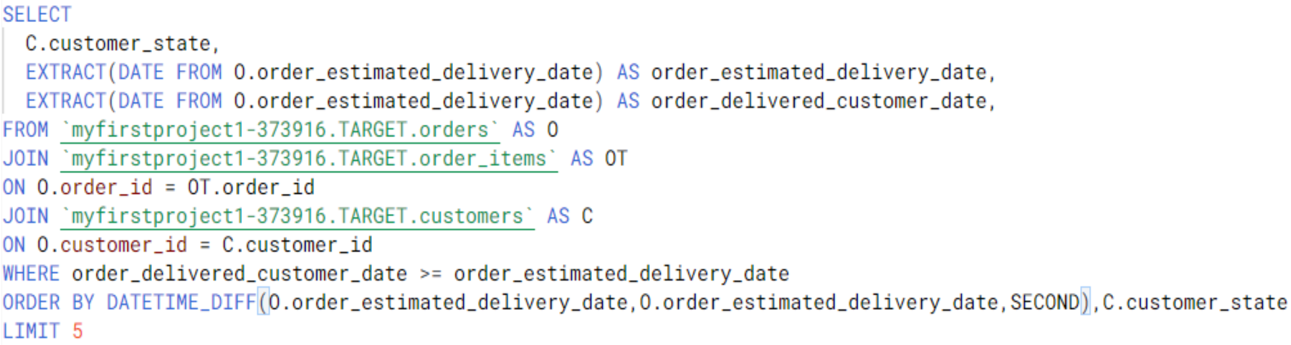
*Fig:* ***QUERY***



| Row | customer_state | AVG_time_to_delivery |
|---|---|---|
| 1 | SP | 8.66 |
| 2 | PR | 11.89 |
| 3 | MG | 11.92 |
| 4 | DF | 12.89 |
| 5 | SC | 14.95 |

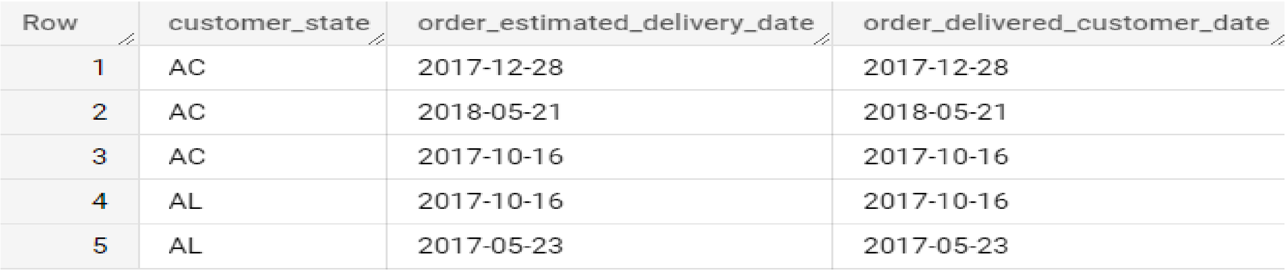
*Fig*: ***QUERY_RESULT***

## 5.7 Top 5 states where delivery is really fast/ not so fast compared to estimated date



```
SELECT
  C.customer_state,
  EXTRACT(DATE FROM O.order_estimated_delivery_date) AS order_estimated_delivery_date,
  EXTRACT(DATE FROM O.order_estimated_delivery_date) AS order_delivered_customer_date,
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.order_items` AS OT
ON O.order_id = OT.order_id
JOIN `myfirstproject1-373916.TARGET.customers` AS C
ON O.customer_id = C.customer_id
WHERE order_delivered_customer_date >= order_estimated_delivery_date
ORDER BY DATETIME_DIFF(O.order_estimated_delivery_date,O.order_estimated_delivery_date,SECOND),C.customer_state
LIMIT 5
```

*Fig:* ***QUERY***

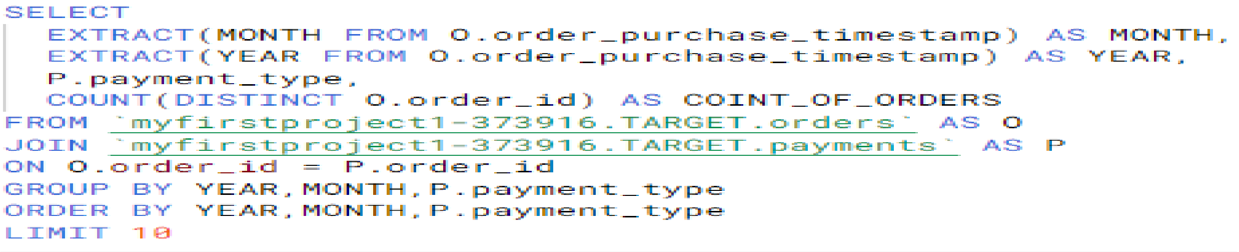


| Row | customer_state | order_estimated_delivery_date | order_delivered_customer_date |
|---|---|---|---|
| 1 | AC | 2017-12-28 | 2017-12-28 |
| 2 | AC | 2018-05-21 | 2018-05-21 |
| 3 | AC | 2017-10-16 | 2017-10-16 |
| 4 | AL | 2017-10-16 | 2017-10-16 |
| 5 | AL | 2017-05-23 | 2017-05-23 |

*Fig*: ***QUERY_RESULT***

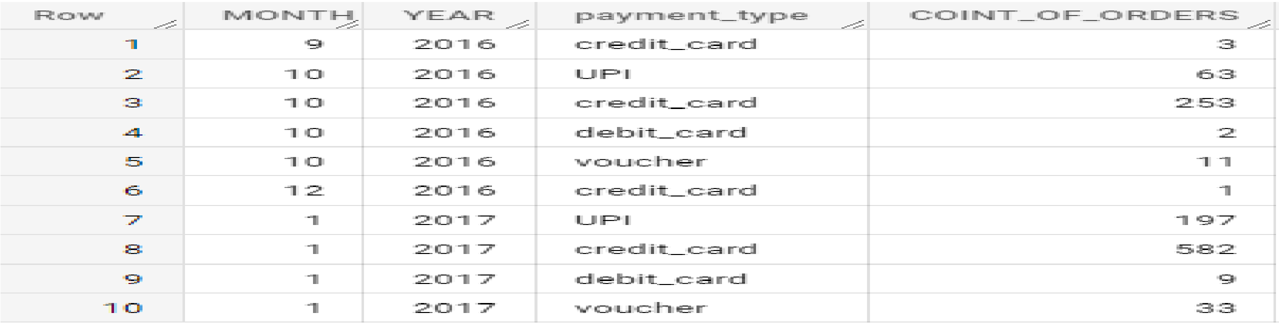
# 6 PAYMENT TYPE ANALYSIS:

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



```
SELECT
    EXTRACT(MONTH FROM O.order_purchase_timestamp) AS MONTH,
    EXTRACT(YEAR FROM O.order_purchase_timestamp) AS YEAR,
    P.payment_type,
    COUNT(DISTINCT O.order_id) AS COINT_OF_ORDERS
FROM `myfirstproject1-373916.TARGET.orders` AS O
JOIN `myfirstproject1-373916.TARGET.payments` AS P
ON O.order_id = P.order_id
GROUP BY YEAR,MONTH,P.payment_type
ORDER BY YEAR,MONTH,P.payment_type
LIMIT 10
```

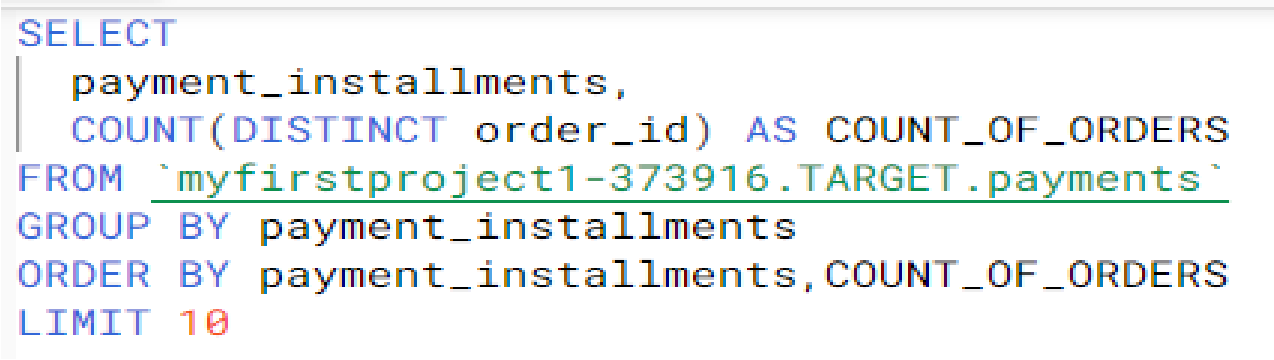
*Fig:* ***QUERY***



| Row | MONTH | YEAR | payment_type | COINT_OF_ORDERS |
|---|---|---|---|---|
| 1 | 9 | 2016 | credit_card | 3 |
| 2 | 10 | 2016 | UPI | 63 |
| 3 | 10 | 2016 | credit_card | 253 |
| 4 | 10 | 2016 | debit_card | 2 |
| 5 | 10 | 2016 | voucher | 11 |
| 6 | 12 | 2016 | credit_card | 1 |
| 7 | 1 | 2017 | UPI | 197 |
| 8 | 1 | 2017 | credit_card | 582 |
| 9 | 1 | 2017 | debit_card | 9 |
| 10 | 1 | 2017 | voucher | 33 |

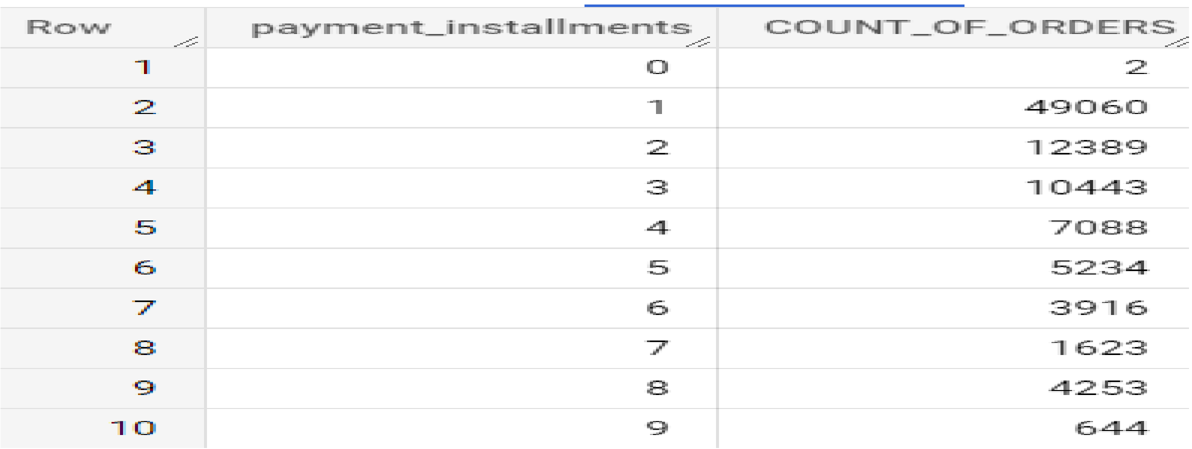
*Fig*: ***QUERY_RESULT***

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



```
SELECT
    payment_installments,
    COUNT(DISTINCT order_id) AS COUNT_OF_ORDERS
FROM `myfirstproject1-373916.TARGET.payments`
GROUP BY payment_installments
ORDER BY payment_installments,COUNT_OF_ORDERS
LIMIT 10
```

*Fig:* ***QUERY***



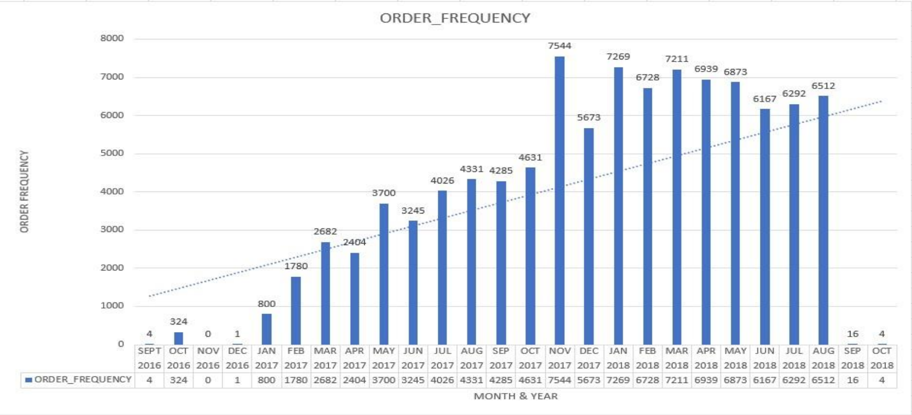
| Row | payment_installments | COUNT_OF_ORDERS |
|---|---|---|
| 1 | 0 | 2 |
| 2 | 1 | 49060 |
| 3 | 2 | 12389 |
| 4 | 3 | 10443 |
| 5 | 4 | 7088 |
| 6 | 5 | 5234 |
| 7 | 6 | 3916 |
| 8 | 7 | 1623 |
| 9 | 8 | 4253 |
| 10 | 9 | 644 |

*Fig*: ***QUERY_RESULT***

# 7 ACTIONABLE INSIGHTS

## 7.1 SEEING TREND IN ORDER OVER MONTH

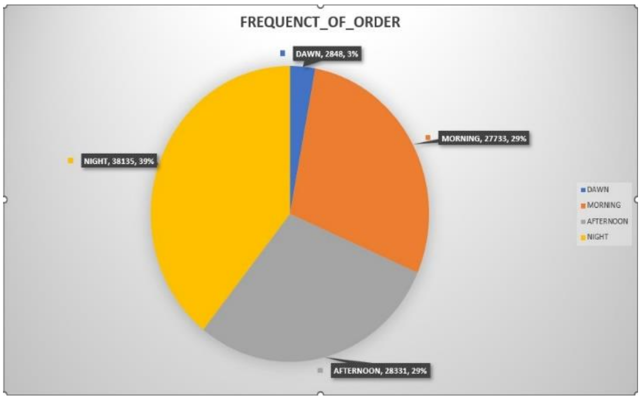
- FROM THE ABOVE GRAPH IT CAN WE SAY THAT NO OF ORDERS ARE MORE INSIDE THE 3rd QUARTILE & MINIMUM IN 4th QUARTILE.
- OVER-ALL THERE IS AN INCREASE IN FREQUENCY OF ORDER OVER A PERIOD OF TIME



| | SEPT 2016 | OCT 2016 | NOV 2016 | DEC 2016 | JAN 2017 | FEB 2017 | MAR 2017 | APR 2017 | MAY 2017 | JUN 2017 | JUL 2017 | AUG 2017 | SEP 2017 | OCT 2017 | NOV 2017 | DEC 2017 | JAN 2018 | FEB 2018 | MAR 2018 | APR 2018 | MAY 2018 | JUN 2018 | JUL 2018 | AUG 2018 | SEP 2018 | OCT 2018 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ■ ORDER_FREQUENCY | 4 | 324 | 0 | 1 | 800 | 1780 | 2682 | 2404 | 3700 | 3245 | 4026 | 4331 | 4285 | 4631 | 7544 | 5673 | 7269 | 6728 | 7211 | 6939 | 6873 | 6167 | 6292 | 6512 | 16 | 4 |

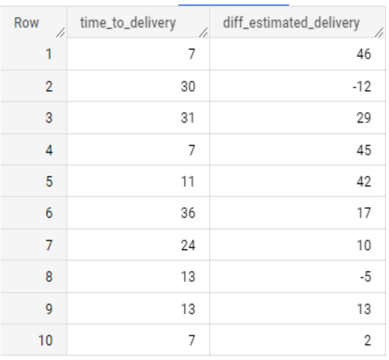
## 7.2 FREQUENCY OF ORDER DURING ENTIRE DAY & NIGHT

- PEOPLE PREFER TO ORDER DURING AFTERNOON FOLLOWED BY NIGHT & MORNING
- FREQUENCY OF ORDER IS MINIMUM DURING DAWN



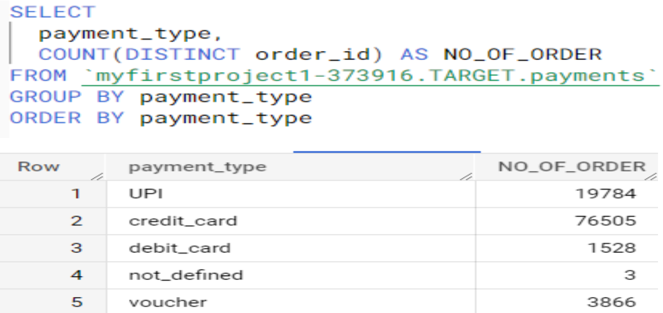
## 7.3 INSIGHT FROM ESTIMATED DELIVERY DATE

- DIFF_ESTIMATED_DELIVERY DATE INDICATE THE ESTIMATED DATE & DELIVERY DATE DIFFERENCE
- IT SHOULD BE MINIMUM AS POSSIBLE FOR CUSTOMER SATISFACTION
- -VE VALUE INDICATE ORDER IS DELIVERED BEFORE THE ESTIMATED TIME WHICH IS PRETTTY GOOD INDICATION FOR CUSTOMER



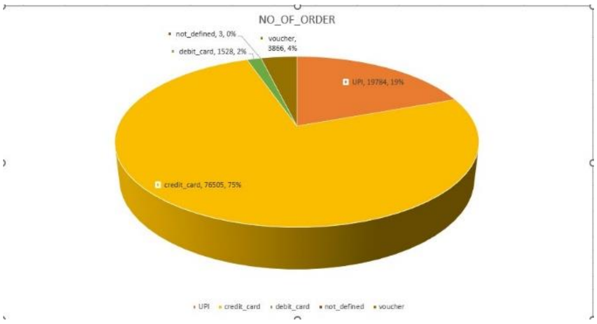
| Row | time_to_delivery | diff_estimated_delivery |
|---|---|---|
| 1 | 7 | 46 |
| 2 | 30 | -12 |
| 3 | 31 | 29 |
| 4 | 7 | 45 |
| 5 | 11 | 42 |
| 6 | 36 | 17 |
| 7 | 24 | 10 |
| 8 | 13 | -5 |
| 9 | 13 | 13 |
| 10 | 7 | 2 |

## 7.4 MODE OF PAYMENT



```
SELECT
  payment_type,
  COUNT(DISTINCT order_id) AS NO_OF_ORDER
FROM `myfirstproject1-373916.TARGET.payments`
GROUP BY payment_type
ORDER BY payment_type
```

| Row | payment_type | NO_OF_ORDER |
|---|---|---|
| 1 | UPI | 19784 |
| 2 | credit_card | 76505 |
| 3 | debit_card | 1528 |
| 4 | not_defined | 3 |
| 5 | voucher | 3866 |



- PEOPLE TEND TO PURCHASE MORE ON CREDIT-CARD & UPI
- PEOPLE GENERALLY AVOID CASH(NOD_DEFINED) FOR PAYMENT

# 8 RECOMMENDATIONS

- SINCE THE NO OF ORDERS IS HIGHEST IN 3RD QUARTER ITS BETTER TO ADVERTISE DURING THIS QUARTER.
- ALSO PROVIDE SOME DISCOUNT/OFFER DURING OTHER QUARTER TO MOTIVATE CUSTOMERS TO PURCHASE MORE & MORE TO INCREASE NO OF ORDER DURING OTHER QUARTER ALSO.
- SINCE PEOPLE PREFER TO PURCHASE DURING EVENING & NIGHT ITS BETTER TO KEEP MORE LABOUR DURING THIS PERIOD TO MAKE THE FLOW EFFICIENT FOR CUSTOMERS AND INCREASE THE PURCHASE EXPERIENCE.
- LABOUR DURING DAWN & MORNING TIME CAN BE REDUCED AND CAN WE SHIFTED IN OTHER TIME SLOT TO FULFILL THE REQUIREMENT.
- ITS BETTER TO REDUCE ESTIMATED TIME-DIFFERENCE TO INCREASE SATISFACTION OF CUSTOMER DURING THEIR PURCHASE. ITS BETTER TO KEEP IT -VE OR MINIMUM FOR BETTER EXPERIENCE OF CUSTOMER.
- PEOPLE PREFER TO BUY USING CREDIT-CARD & UPI. SO ITS BETTER TO PROVIDE SOME DISCOUNT OR OFFER & ALSO SUPPORT DIVERSITY OF CREDIT-CARD FOR TRANSACTIONS