**TARGET**

**SQL Business Case Study**

* Himkant Nigam

1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
   1. **Data type of all columns in the "customers" table.**

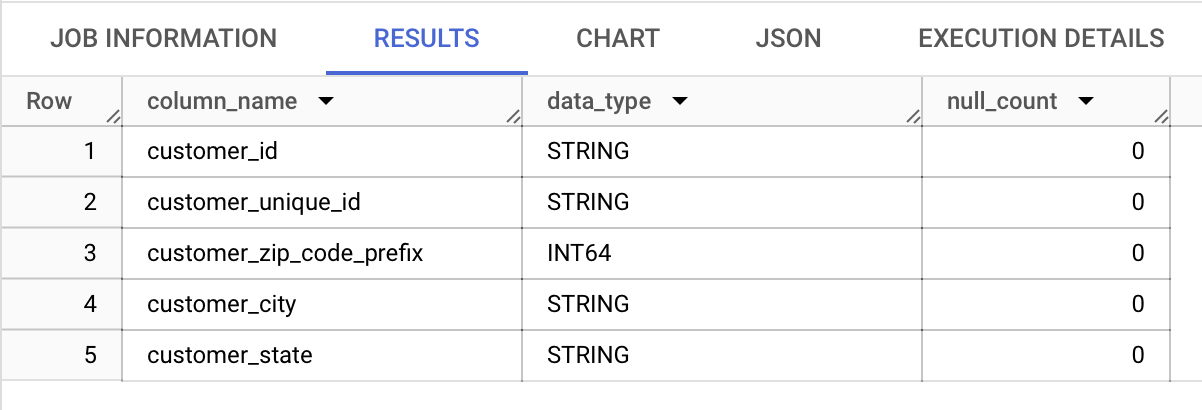
SELECT column\_name, data\_type,

SUM(CASE WHEN column\_name IS NULL THEN 1 ELSE 0 END) AS null\_count

FROM sql123-408717.sql\_buzcase.INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name = 'customers'

GROUP BY column\_name, data\_type;



**INSIGHTS** : There are 5 columns as displayed above in which only customer zip code column is integer rest all are string with no NULL values in any column.

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

SELECT MIN(order\_purchase\_timestamp) AS min\_order\_timestamp, MAX(order\_purchase\_timestamp) AS max\_order\_timestamp

FROM sql123-408717.sql\_buzcase.orders;

A screenshot of a chart

Description automatically generated

**INSIGHTS** : The data of orders table is between 2016-09-04 21:15:19 UTC and 2018-10-17 17:30:18 UTC.

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

SELECT customer\_city, customer\_state, COUNT(\*) AS order\_count

FROM sql123-408717.sql\_buzcase.customers AS c

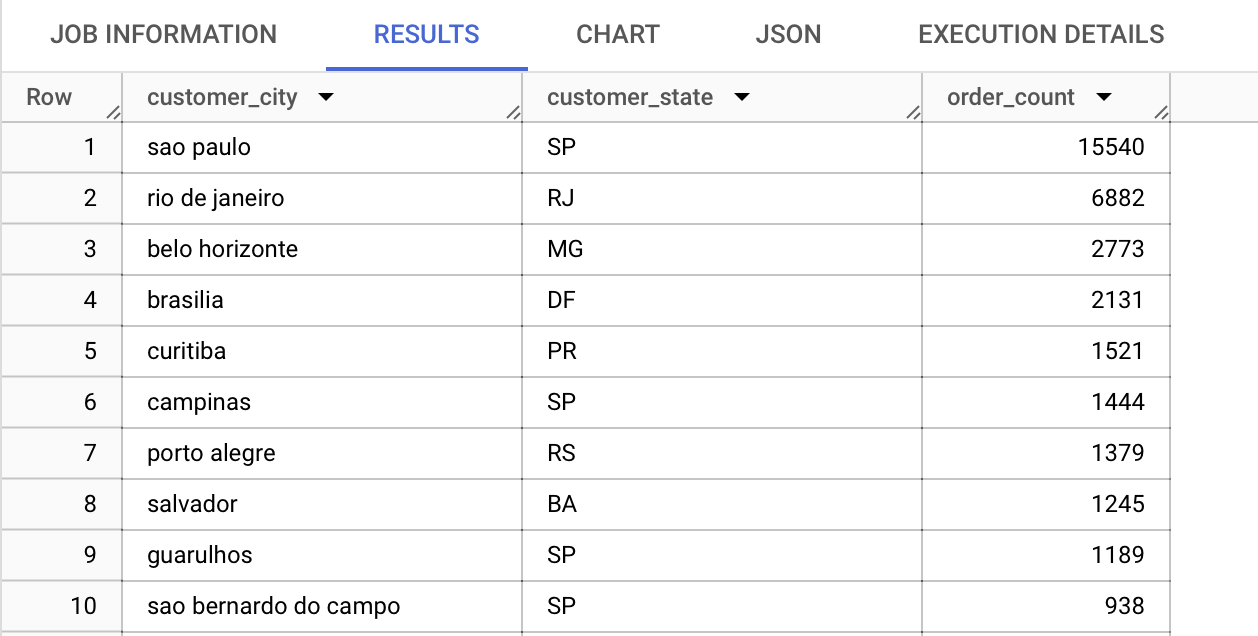
JOIN sql123-408717.sql\_buzcase.orders AS o

ON c.customer\_id = o.customer\_id

WHERE o.order\_purchase\_timestamp BETWEEN (SELECT MIN(order\_purchase\_timestamp) FROM sql123-408717.sql\_buzcase.orders) AND (SELECT MAX(order\_purchase\_timestamp) FROM sql123-408717.sql\_buzcase.orders)

GROUP BY customer\_city, customer\_state

ORDER BY order\_count DESC;



**INSIGHTS** : - SP state is coming most of the time in top 10 orders while the order count is arranged in descending order.

* Sao Paulo contains most of the orders which is almost twice the count of 2nd highest count in Rio de Janerio.
* 2nd highest count is also almost twice of 3rd highest order count.
* There are 1260 cities that contain 1 order.
* There are 703 cities that contain 2 orders.
* There are 441 cities that contain 3 orders.
* There are 291 cities that contain 4 orders.
* There are 210 cities that contain 5 orders and so on.

**RECOMMENDATION** :

* We can focus on 2nd an 3rd city for increasing orders in order to level up or compete with 1st city. This could have been done with some promotional schemes. Also, we can increase the customers in the city by offering some subsidy from government as increase in customers leads to more number of orders.

1. **In-depth Exploration:**
   1. **Is there a growing trend in the no. of orders placed over the past years?**

SELECT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year, COUNT(\*) AS order\_count

FROM sql123-408717.sql\_buzcase.orders

GROUP BY year

ORDER BY year;

A screenshot of a computer

Description automatically generated

**INSIGHTS** :

* As we can see from the data, ***YES*** there is a growing trend year over year as the orders are increasing every year.
* There is an almost 20% increase in orders from 2017 to 2018.

**RECOMMENDATION** :

* N/A
  1. **Can we see some kind of monthly seasonality in terms of the no. of orders being placed?**

SELECT EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month, COUNT(\*) AS order\_count

FROM `sql123-408717.sql\_buzcase.orders`

GROUP BY month

ORDER BY month;

**A screenshot of a table

Description automatically generated**

**INSIGHTS :**

* Least orders are towards the end of year with least order in 9th month.
* Highest orders are in 5th , 7th and 8th month.
* Orders are consistent in the starting of year.

**RECOMMENDATION** :

* We can give promotional offers in particular months like last 4 months so as to increase orders from customers.

**#this question can be further answered in terms of quarters**

WITH monthly\_orders AS (

SELECT

EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

COUNT(\*) AS order\_count

FROM sql123-408717.sql\_buzcase.orders

GROUP BY month ),

quarter\_orders AS (

SELECT month, order\_count,

NTILE(4) OVER (ORDER BY month) AS quarter

FROM monthly\_orders )

SELECT month, order\_count, quarter,

SUM(order\_count) OVER (PARTITION BY quarter) AS quarter\_cumulative

FROM quarter\_orders

ORDER BY month;

A screenshot of a spreadsheet

Description automatically generated

**INSIGHTS :**

* 4th quarter contains the least number of orders.
* 2nd quarter contains the highest number of orders.
* There is increasing trend in order count in 1st quarter.
* Both highest and lowest number of orders are in 3rd quarter.

**RECOMMENDATION** :

* We can give promotional offers in particular months like 4th quarter so as to increase orders from customers.
  1. **During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)**
     + - **0-6 hrs : Dawn**
       - **7-12 hrs : Mornings**
       - **13-18 hrs : Afternoon**
       - **19-23 hrs : Night**

SELECT

CASE

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 7 AND 12 THEN 'Morning'

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'

ELSE 'Night'

END AS order\_time\_period,

COUNT(\*) AS order\_count

FROM `sql123-408717.sql\_buzcase.orders` AS o

JOIN `sql123-408717.sql\_buzcase.customers` AS c

ON o.customer\_id = c.customer\_id

GROUP BY order\_time\_period

ORDER BY order\_count DESC;

A screenshot of a computer

Description automatically generated

**INSIGHTS** :

* AFTERNOON time has maximum orders.
* DAWN time has least orders.
* Morning and Night orders are almost equivalent.

**RECOMMENDATION** :

* We can give attention to customers who are ordering more frequently by giving some kind of reminders or offers at DAWN time.

1. **Evolution of E-commerce orders in the Brazil region:**
   1. **Get the month-on-month no. of orders placed in each state.**

SELECT

EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

customer\_state,

COUNT(\*) AS order\_count

FROM sql123-408717.sql\_buzcase.orders AS o

JOIN sql123-408717.sql\_buzcase.customers AS c

ON o.customer\_id = c.customer\_id

GROUP BY month, customer\_state

ORDER BY month, order\_count DESC;

A screenshot of a computer

Description automatically generated

**INFERENCE :**

* Each month has 27 records stating that there are 27 different states in Brazil.
* SP state contains the maximum number of orders for 1st month which we also saw in question 1.3 for highest number of order count in given time period.
* SP state has least orders in last 4 months of the year.

**RECOMMENDATION** : N/A

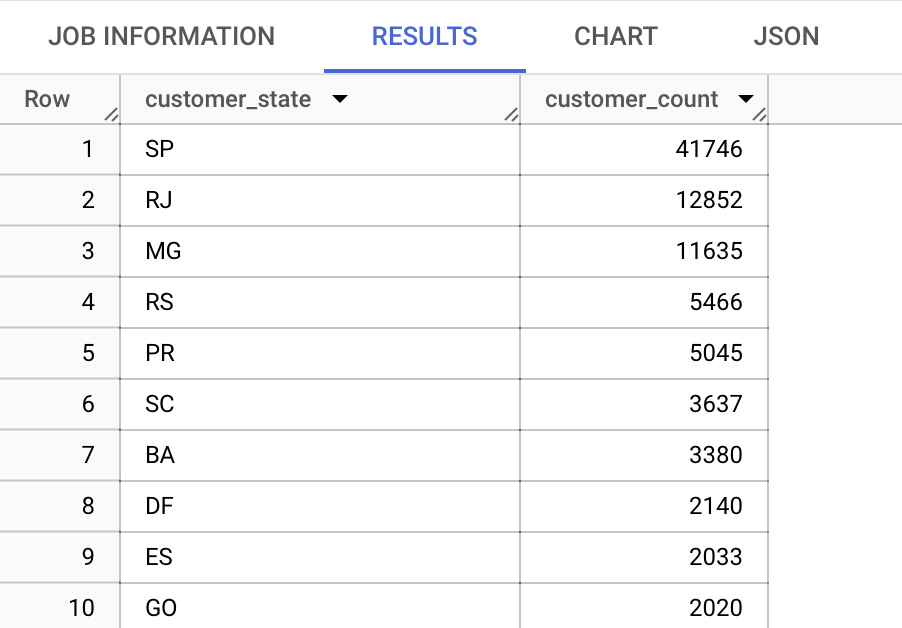
* 1. **How are the customers distributed across all the states?**

SELECT customer\_state, COUNT(DISTINCT customer\_id) AS customer\_count

FROM sql123-408717.sql\_buzcase.customers

GROUP BY customer\_state

ORDER BY customer\_count DESC;



**INFERENCE** :

* Most number of customers are in SP which are almost 3x of RJ(2nd highest numbers of customers) and MG (3rd highest number of customers).
* RR has least number of customers (46) followed by AP(68) then AC(81).

**RECOMMENDATION** :

* In top 3 states we have most of the customers we can focus more on strategizing offers and prices to gain maximum orders and hence revenue.

1. **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**
   1. **Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).**

WITH payment\_years AS (

SELECT

EXTRACT(YEAR FROM order\_purchase\_timestamp) AS order\_year,

EXTRACT(MONTH FROM order\_purchase\_timestamp) AS order\_month,

SUM(payment\_value) AS total\_payment

FROM sql123-408717.sql\_buzcase.payments AS p

JOIN sql123-408717.sql\_buzcase.orders AS o

USING(order\_id)

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) IN (2017, 2018)

AND EXTRACT(MONTH FROM order\_purchase\_timestamp) BETWEEN 1 AND 8

GROUP BY order\_year, order\_month

),

yearly\_totals AS (

SELECT order\_year,

SUM(total\_payment) AS total\_payment\_year

FROM payment\_years

GROUP BY order\_year

)

SELECT

(yt2018.total\_payment\_year - yt2017.total\_payment\_year) / yt2017.total\_payment\_year \* 100 AS percent\_increase

FROM yearly\_totals yt2017

JOIN yearly\_totals yt2018 ON yt2017.order\_year = 2017 AND yt2018.order\_year = 2018;

A screenshot of a computer screen

Description automatically generated

**INFERENCE** : There is a change of almost 137% in the cost of orders from year 2017 to 2018 which includes months between Jan to Aug only.

**RECOMMENDATION** : N/A

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

SELECT customer\_state,

SUM(price) AS total\_order\_price,

AVG(price) AS average\_order\_price

FROM sql123-408717.sql\_buzcase.customers AS c

LEFT JOIN sql123-408717.sql\_buzcase.orders AS o

USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi

USING(order\_id)

GROUP BY customer\_state

ORDER BY total\_order\_price DESC, average\_order\_price DESC;

A screenshot of a computer

Description automatically generated

**INFERENCE** :

* SP state has maximum total order price with least average order price, so we might say that either number of items per order are comparatively less or the orders contain items which are comparatively cheap.
* As we have seen this order of SP > RJ > MG, it gets proved by the number of unique customers in these states.
* State PB has highest average order price followed by AL and AC states stating that the customer order in bulk rather than ordering on daily basis.
* State RR has least total order price followed by AP and AC states.

**RECOMMENDATION :**

* Since the total order price is maximum for SP but average order price is minimum we can offer them clubbed offers to increase the average order price, for example 1L cold drink costs 60 INR but 3L costs 150 INR, this will increase the total cost as well as average cost.
  1. **Calculate the Total & Average value of order freight for each state.**

SELECT customer\_state,

SUM(freight\_value) AS total\_freight\_value,

AVG(freight\_value) AS average\_freight\_value

FROM sql123-408717.sql\_buzcase.customers AS c

LEFT JOIN sql123-408717.sql\_buzcase.orders AS o

USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi

USING(order\_id)

GROUP BY customer\_state

ORDER BY customer\_state;

A screenshot of a data

Description automatically generated

**INFERENCE** :

* SP state has maximum total Freight value with least average freight value, so we might say that either number of items per order are comparatively less or the orders contain items which are comparatively cheap.
* As we have seen this order of SP > RJ > MG, it gets proved by the number of unique customers in these states.
* State RR has highest average freight value followed by PB and RO states stating that the customer order in bulk rather than ordering on daily basis.
* State RR has least total freight value followed by AP and AC states.

**RECOMMENDATION :**

* Since the total freight value is maximum for SP but average freight value is minimum we can offer them clubbed offers to increase the average order price, for example 1L cold drink costs 60 INR but 3L costs 150 INR, this will increase the total freight value as well as average freight value.

1. **Analysis based on sales, freight and delivery time.**
   1. **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.  
      You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:**
      * + **time\_to\_deliver = order\_delivered\_customer\_date - order\_purchase\_timestamp**
        + **diff\_estimated\_delivery = order\_delivered\_customer\_date - order\_estimated\_delivery\_date**

SELECT order\_id, DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY) AS delivery\_time,

DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date, DAY) AS diff\_estimated\_delivery

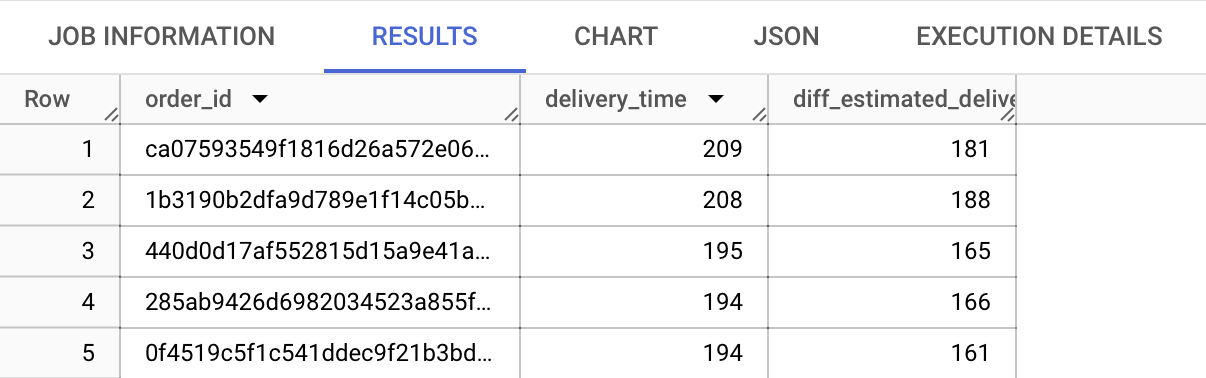
FROM sql123-408717.sql\_buzcase.orders

WHERE order\_delivered\_customer\_date IS NOT NULL

AND order\_purchase\_timestamp IS NOT NULL

AND order\_estimated\_delivery\_date IS NOT NULL

ORDER BY delivery\_time DESC, diff\_estimated\_delivery DESC;



**INFERENCE** :

* Maximum time to deliver an order was 209 days while minimum was 0 days which means the order was delivered on the same day.
* In case of time between estimated and actual delivery date, the maximum was 27 days(the order was delivered 27 days after the estimated date) while minimum was 146 days(order was delivered 146 days before the estimated date).

**RECOMMENDATION** :

* We should focus more on delivering within time, especially for those orders exceeding 2 or 3 weeks’ time because if someone is ordering something, he should be needing it.
  1. **Find out the top 5 states with the highest & lowest average freight value.**

SELECT customer\_state,

ROUND(AVG(freight\_value),4) AS average\_freight\_value

FROM sql123-408717.sql\_buzcase.customers AS c

JOIN sql123-408717.sql\_buzcase.orders AS o USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi USING(order\_id)

GROUP BY customer\_state

ORDER BY average\_freight\_value DESC

LIMIT 5;



SELECT customer\_state,

ROUND(AVG(freight\_value),4) AS average\_freight\_value

FROM sql123-408717.sql\_buzcase.customers AS c

JOIN sql123-408717.sql\_buzcase.orders AS o

USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi

USING(order\_id)

GROUP BY customer\_state

ORDER BY average\_freight\_value

LIMIT 5;

A screenshot of a computer

Description automatically generated

**INSIGHTS** :

* Top 5 states by average freight value are RR, PB, RO, AC, PI. As seen above these are states with least customers and large/bulk ordering.
* Bottom 5 states by average freight value are SP, PR, MG, RJ, DF. As seen above these are states with maximum customers with large number of orders with less freight value.
  1. **Find out the top 5 states with the highest & lowest average delivery time.**

SELECT customer\_state,ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)),4) AS avg\_delivery\_time\_day

FROM sql123-408717.sql\_buzcase.customers AS c

JOIN sql123-408717.sql\_buzcase.orders AS o USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi USING(order\_id)

WHERE order\_delivered\_customer\_date IS NOT NULL

AND order\_purchase\_timestamp IS NOT NULL

GROUP BY customer\_state

ORDER BY avg\_delivery\_time\_day DESC

LIMIT 5;

A screenshot of a computer

Description automatically generated

SELECT customer\_state,ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)),4) AS avg\_delivery\_time\_day

FROM sql123-408717.sql\_buzcase.customers AS c

JOIN sql123-408717.sql\_buzcase.orders AS o USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi USING(order\_id)

WHERE order\_delivered\_customer\_date IS NOT NULL

AND order\_purchase\_timestamp IS NOT NULL

GROUP BY customer\_state

ORDER BY avg\_delivery\_time\_day

LIMIT 5;

A screenshot of a computer

Description automatically generated

**INSIGHTS** :

* Average delivery time is minimum for state SP (which has highest number of customers and minimum average freight value).
* Average delivery time is maximum for RR (which has maximum average freight value and least customers).
  1. **Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.  
     You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.**

WITH delivery\_difference AS (

SELECT customer\_state, ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date,DAY)),4) AS avg\_delivery\_difference

FROM sql123-408717.sql\_buzcase.customers AS c

JOIN sql123-408717.sql\_buzcase.orders AS o

USING(customer\_id)

JOIN sql123-408717.sql\_buzcase.order\_items AS oi

USING(order\_id)

WHERE order\_delivered\_customer\_date IS NOT NULL

AND order\_estimated\_delivery\_date IS NOT NULL

GROUP BY customer\_state )

SELECT customer\_state, avg\_delivery\_difference

FROM delivery\_difference

ORDER BY avg\_delivery\_difference

LIMIT 5;

A screenshot of a computer

Description automatically generated

**INFERENCE** : All the states that have comparatively faster delivery(delivery before estimated date) have low number of customers.

1. **Analysis based on the payments:**
   1. **Find the month-on-month no. of orders placed using different payment types.**

SELECT

EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year,

EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

payment\_type, COUNT(\*) AS num\_orders

FROM sql123-408717.sql\_buzcase.orders AS o

JOIN sql123-408717.sql\_buzcase.payments AS p USING(order\_id)

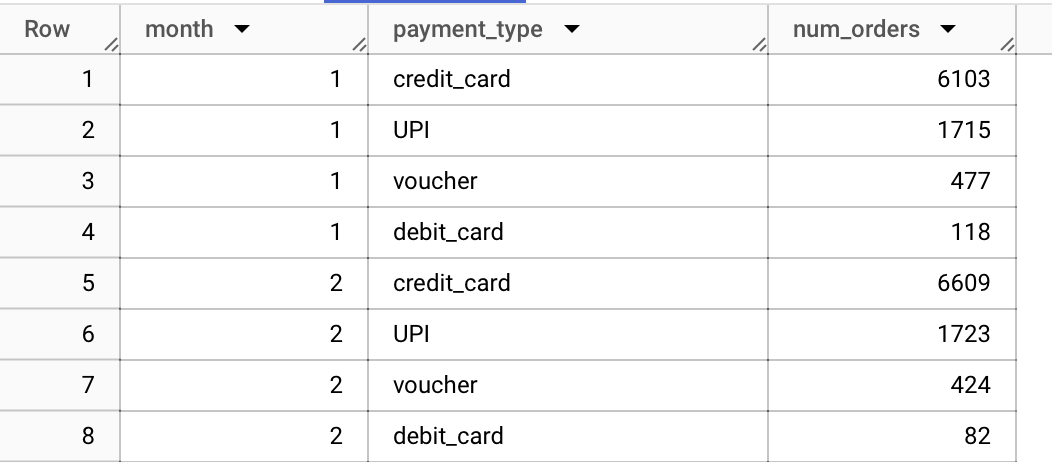
GROUP BY year, month, payment\_type

ORDER BY year, month, num\_orders DESC;

A screenshot of a computer

Description automatically generated

**#in case we remove years for month on month analysis, we get:**



**INSIGHTS** :

* There are no transaction for 2016 November or the data may be missing.
* Maximum transactions are from Credit Card.
* There are 4 types of payments : credit card, UPI, voucher, debit card
* Data starts from 2016 September.
  1. **Find the no. of orders placed on the basis of the payment installments that have been paid.**

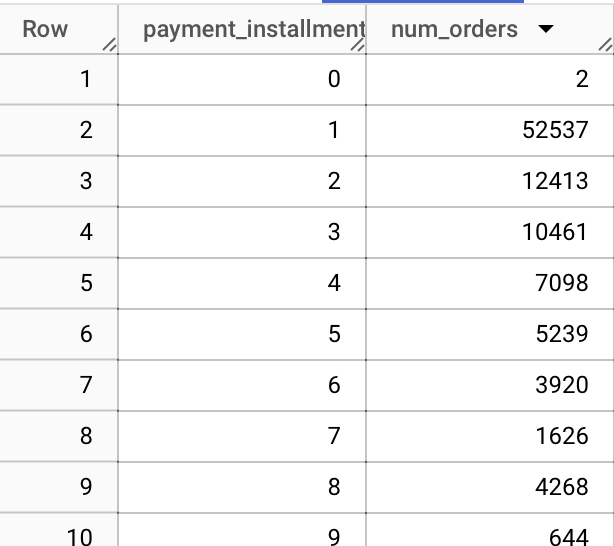
SELECT payment\_installments,

COUNT(\*) AS num\_orders

FROM sql123-408717.sql\_buzcase.payments

WHERE payment\_value!= 0

GROUP BY payment\_installments;



**#for condition where payment value = 0, we get :**

A screenshot of a computer

Description automatically generated

**INSIGHTS** :

* There are total 9 records where payment value = 0 in 1st month.
* There are 2 payments done at 0 , which might mean pre payments.
* Payments count keep reducing as we proceed which might mean that people take installments for less time.
* Also, there is a sudden increase of payment instalment in 8th and 10th month which might mean that most of the instalment periods got over in those months.

THE END