Exploratory Data Analysis (EDA) using SQL (‘Bigquery’) queries.

This business case has information of 100k orders from 2016 to 2018 made at the company in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

OBJECTIVE

To do the analysis on the dataset and gain insights from it and then make recommendations for the company.

In order to do this, we’ll follow the following steps. We’ll use **Bigquery** for this case study.

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
   1. Data type of columns in a table
   2. Time period for which the data is given
   3. Cities and States of customers ordered during the given period
2. In-depth Exploration:
   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 at specific months?
   2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
3. Evolution of E-commerce orders in the Brazil region:
   1. Get month on month orders by states
   2. Distribution of customers across the states in Brazil
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
   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
   2. Mean & Sum of price and freight value by customer state

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
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
3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery
4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
6. Top 5 states with highest/lowest average time to delivery
7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

6. Payment type analysis:

1. Month over Month count of orders for different payment types
2. Count of orders based on the no. of payment installments

**About data:**

*The****customers*** *table contain following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| customer\_id | Id of the consumer who made the purchase. |
| customer\_unique\_id | Unique Id of the consumer. |
| customer\_zip\_code\_prefix | Zip Code of the location of the consumer. |
| customer\_city | Name of the City from where order is made. |
| customer\_state | State Code from where order is made(Ex- sao paulo-SP). |

*The****sellers*** *table contains following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| seller\_id | Unique Id of the seller registered |
| seller\_zip\_code\_prefix | Zip Code of the location of the seller. |
| seller\_city | Name of the City of the seller. |
| seller\_state | State Code (Ex- sao paulo-SP) |

*The****order\_items*** *table contain following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| order\_id | A unique id of order made by the consumers. |
| order\_item\_id | A Unique id given to each item ordered in the order. |
| product\_id | A unique id given to each product available on the site. |
| seller\_id | Unique Id of the seller registered in Target. |
| shipping\_limit\_date | The date before which shipping of the ordered product must be completed. |
| price | Actual price of the products ordered . |
| freight\_value | Price rate at which a product is delivered from one point to another. |

*The****geolocations*** *table contain following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| geolocation\_zip\_code\_prefix | first 5 digits of zip code |
| geolocation\_lat | latitude |
| geolocation\_lng | longitude |
| geolocation\_city | city name |
| geolocation\_state | state |

*The****payments*** *table contain following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| order\_id | A unique id of order made by the consumers. |
| payment\_sequential | sequences of the payments made in case of EMI. |
| payment\_type | mode of payment used.(Ex-Credit Card) |
| payment\_installments | number of installments in case of EMI purchase. |
| payment\_value | Total amount paid for the purchase order. |

*The****orders*** *table contain following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| order\_id | A unique id of order made by the consumers. |
| customer\_id | Id of the consumer who made the purchase. |
| order\_status | status of the order made i.e delivered, shipped etc. |
| order\_purchase\_timestamp | Timestamp of the purchase. |
| order\_delivered\_carrier\_date | delivery date at which carrier made the delivery. |
| order\_delivered\_customer\_date | date at which customer got the product. |
| order\_estimated\_delivery\_date | estimated delivery date of the products. |

*The****reviews*** *table contain following features:*

|  |  |
| --- | --- |
| **Features** | **Description** |
| review\_id | Id of the review given on the product ordered by the order id. |
| order\_id | A unique id of order made by the consumers. |
| review\_score | review score given by the customer for each order on the scale of 1–5. |
| review\_comment\_title | Title of the review |
| review\_comment\_message | Review comments posted by the consumer for each order. |
| review\_creation\_date | Timestamp of the review when it is created. |
| review\_answer\_timestamp | Timestamp of the review answered. |

*The****products*** *table contain following features:*

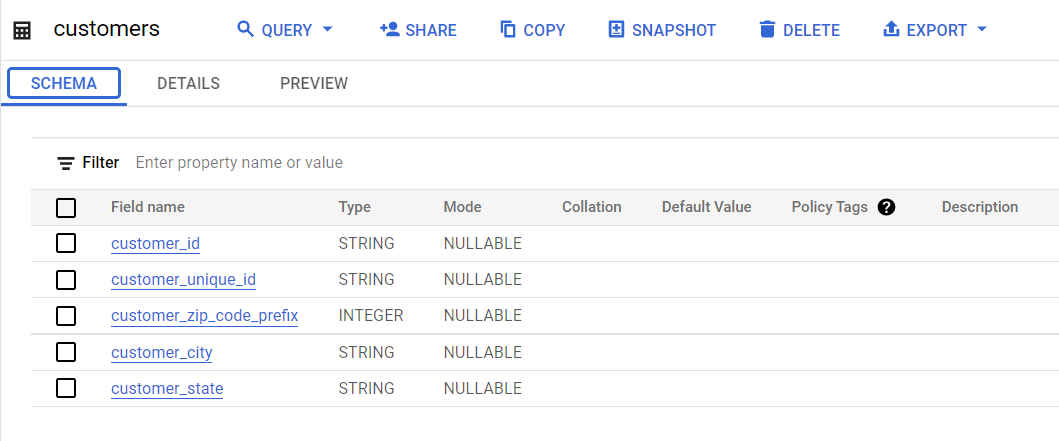
|  |  |
| --- | --- |
| **Features** | **Description** |
| product\_id | A unique identifier for the proposed project. |
| product\_category\_name | Name of the product category |
| product\_name\_lenght | length of the string which specifies the name given to the products ordered. |
| product\_description\_lenght | length of the description written for each product ordered on the site. |
| product\_photos\_qty | Number of photos of each product ordered available on the shopping portal. |
| product\_weight\_g | Weight of the products ordered in grams. |
| product\_length\_cm | Length of the products ordered in centimeters. |
| product\_height\_cm | Height of the products ordered in centimeters. |
| product\_width\_cm | width of the product ordered in centimeters. |

**High level overview of relationship between datasets:**

Diagram

Description automatically generated

1. **Checking the structure and characteristic of the dataset provided (of ‘Retail\_company’).**
   1. *Datatypes* of the columns of the tables:
      1. CUSTOMERS



Number of rows: 99,441

Count of unique customers, cities, and states:

Query

SELECT

COUNT(DISTINCT customer\_id) AS no\_of\_unique\_customers,

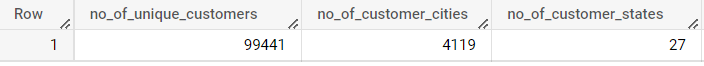
COUNT(DISTINCT customer\_city)AS no\_of\_customer\_cities,

COUNT(DISTINCT customer\_state) AS no\_of\_customer\_states

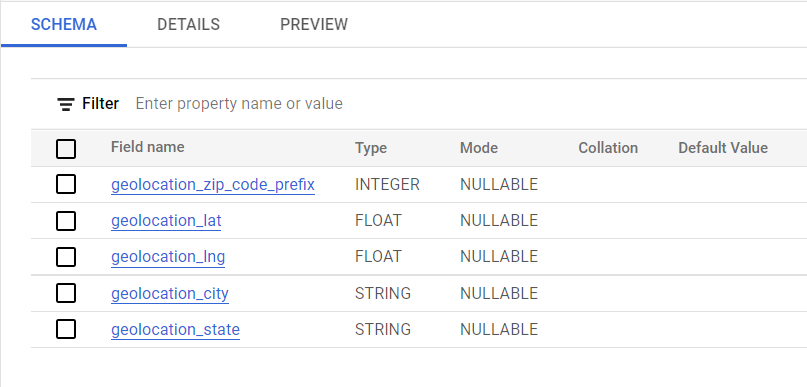
FROM

`scalers-sql.Retail\_company.customers`

Result



* + 1. GEOLOCATION



Number of rows: 1,000,163

Count of total unique cities and states available:

Query

SELECT

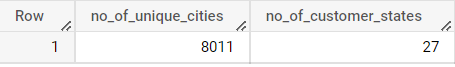
COUNT(DISTINCT geolocation\_city) AS no\_of\_unique\_cities,

COUNT(DISTINCT geolocation\_state)AS no\_of\_customer\_states

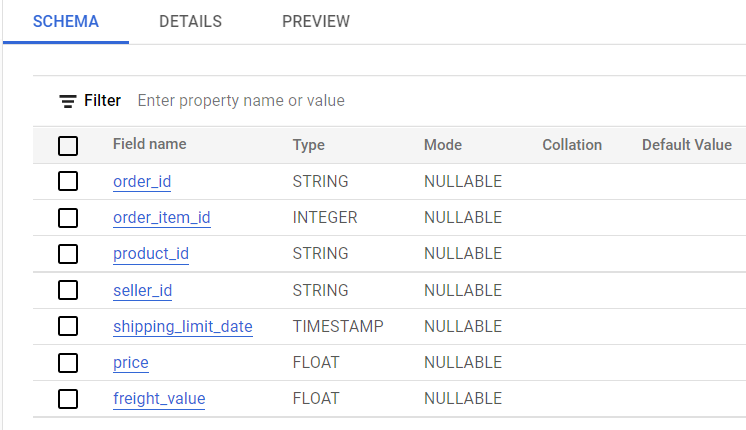
FROM

`scalers-sql.Retail\_company.geolocation`

Result



* + 1. ORDER\_ITEMS



Number of rows: 112,650

Count of total unique orders, sellers and products:

Query

SELECT

COUNT(DISTINCT order\_id) AS no\_of\_orders,

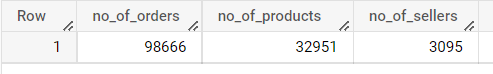
COUNT(DISTINCT product\_id)AS no\_of\_products,

COUNT(DISTINCT seller\_id)AS no\_of\_sellers

FROM

`scalers-sql.Retail\_company.order\_items`

Result



*Years* from the product delivery date:

Query

SELECT

DISTINCT EXTRACT(year

FROM

shipping\_limit\_date) AS years

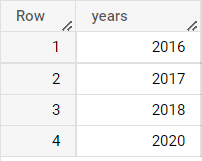
FROM

`scalers-sql.Retail\_company.order\_items`

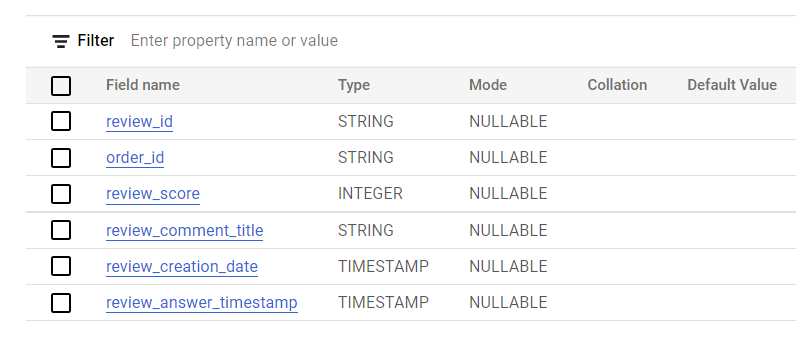
ORDER BY

years

Result

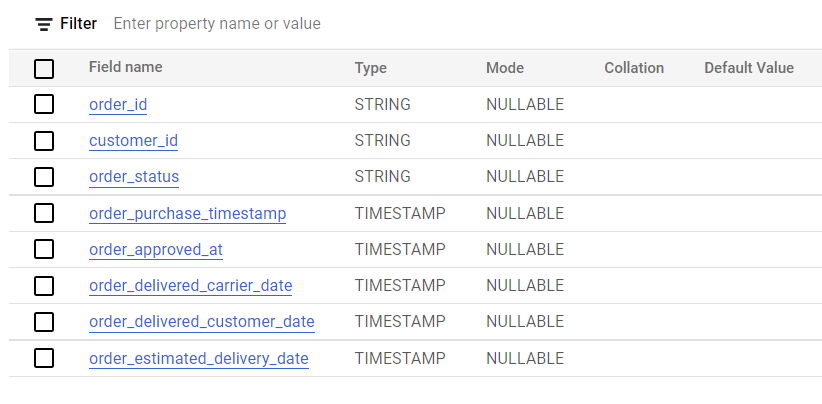


* + 1. ORDER\_REVIEWS



Number of rows: 99,224

* + 1. ORDERS



Number of rows: 99,441

Count of unique orders,products and sellers:

Query

SELECT

COUNT(DISTINCT order\_id) AS No\_of\_orders,

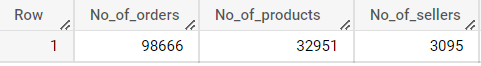
COUNT(DISTINCT product\_id) AS No\_of\_products,

COUNT(DISTINCT seller\_id) AS No\_of\_sellers

FROM

`scalers-sql.Retail\_company.order\_items`

Result



*Time-period* for ‘orders’ table are:

Query

SELECT

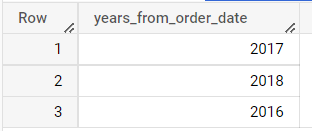
DISTINCT EXTRACT(year FROM order\_purchase\_timestamp) AS years\_from\_order\_date,

-- DISTINCT EXTRACT(year FROM order\_delivered\_customer\_date) AS years\_from\_deivered\_date

FROM

`scalers-sql.Retail\_company.orders`

Result



*Time-periods* this data is associated:

Query

SELECT

DISTINCT EXTRACT(year FROM order\_approved\_at) AS years\_from\_approved\_order\_date,

FROM

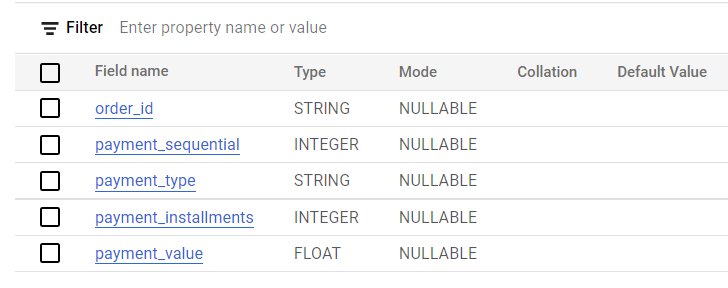
`scalers-sql.Retail\_company.orders`

Results



*This implies that there are some null values associated with this column(approved\_order\_date). This signifies that there are some orders for which year of approval for order placement is not decided.*

* + 1. PAYMENTS



Number of rows: 103,886

Type of payments:

Query

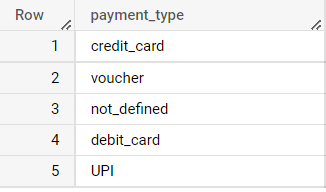
SELECT

DISTINCT payment\_type

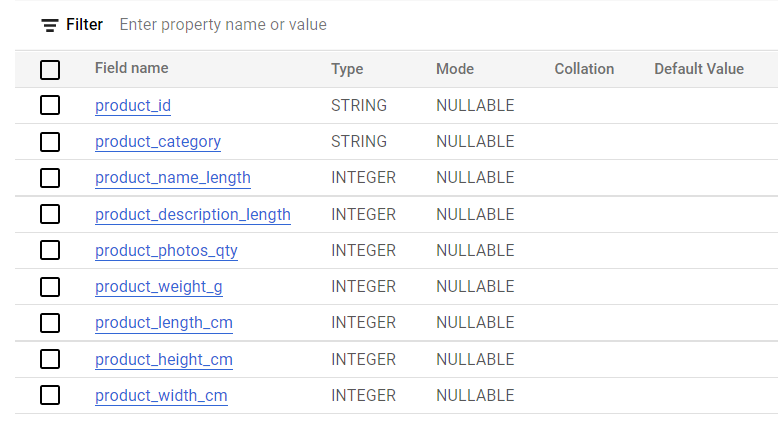
FROM

`scalers-sql.Retail\_company.payments`

Result



* + 1. PRODUCTS



Number of rows: 32,951

Count of distinct products and product categories:

Query

SELECT

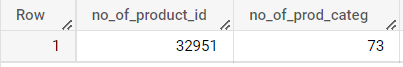
COUNT(DISTINCT product\_id) AS no\_of\_product\_id,

COUNT(DISTINCT product\_category)AS no\_of\_prod\_categ

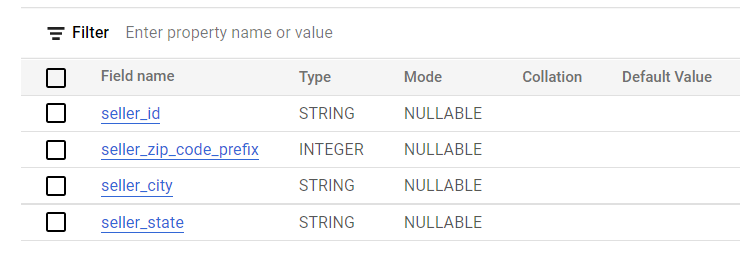
FROM

`scalers-sql.Retail\_company.products`

Result



* + 1. SELLERS



Number of rows: 3,095

Count of unique sellers:

Query

SELECT

COUNT(DISTINCT seller\_id) AS no\_of\_sellers,

COUNT(DISTINCT seller\_city)AS no\_of\_seller\_cities,

COUNT(DISTINCT seller\_state) AS no\_of\_seller\_states

FROM

`scalers-sql.Retail\_company.sellers`

Result



1. **In-depth exploration**
   1. Is there a growing trend on e-com in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at months?

In order to check if e-com business is growing in Brazil we can check *the % change of the orders placed per year*.

Query

SELECT

COUNT(order\_id) AS no\_of\_orders,

order\_year

FROM (

SELECT \*,

EXTRACT(year FROM order\_purchase\_timestamp) AS order\_year

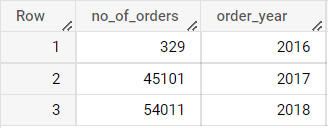
FROM `scalers-sql.Retail\_company.orders`

)

GROUP BY order\_year

ORDER BY order\_year

Results



As per the above table, approx. **8.96%** growth was observed in

order-placement in ‘Retail\_company’ from 2017 to 2018.

Now in order to check the ‘*order placed per month’* by the customers:

Query

SELECT

order\_month,

COUNT(order\_id) AS order\_count

FROM (

SELECT

order\_id,

EXTRACT (month

FROM

order\_purchase\_timestamp) AS order\_month

FROM

`scalers-sql.Retail\_company.orders` ) AS x

GROUP BY

order\_month

ORDER BY

order\_count

Result



In the months of **8***(August)* maximum number of orders were placed followed by month **5***(May)* and **7***(July)*. And in the month of **9***(September)* the minimum number of orders were placed followed by the month **10** *(October)* and **12** *(December)*. In Brazil, *summer* is December through March and *winter* June through September, *autumn* through March to May. So, it can be said that Brazilians like to shop in the months of winters and the start of autumn.

Next, to know at what *hour or time of the day most orders are placed*:

Query

SELECT

order\_hour,

COUNT(order\_id) AS orders\_count

FROM (

SELECT

order\_id,

EXTRACT(hour

FROM

order\_purchase\_timestamp) AS order\_hour

FROM

`scalers-sql.Retail\_company.orders`) AS x

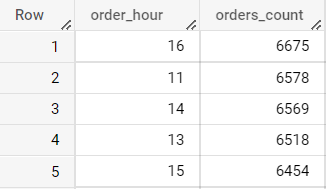
GROUP BY

order\_hour

ORDER BY

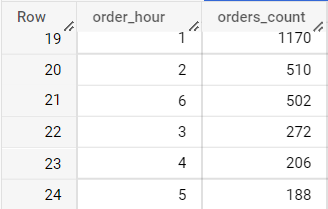
orders\_count DESC

Result



At the **16th** hour of the day *max number of orders were placed* by the customers. However, it can also be observed there is *not a very significant difference for the top 4 hours of the day* i.e. 16th,11th,14th and 13th hour in terms of order number.

*Least* number of orders were placed in the following hours:



Thus, the *least* number of orders get placed in the *night* time.

* 1. What time do Brazilian customers tend to buy (Dawn,Morning, Afternoon and Night)

|  |  |
| --- | --- |
| Time of the day (in 24 hour clock) | Category (as per day time) |
| [04:00:00 - 06:00:00) | Dawn |
| [6:00:00 - 12:00:00) | Morning |
| [12:00:00 - 16:00:00) | Afternoon |
| [16:00:00 - 19:00:00) | Evening |
| Remaining | Night |

Query

SELECT

COUNT(order\_id) AS orders\_count,

CASE

WHEN order\_time >= '6:00:00' AND order\_time < "12:00:00" THEN 'Morning'

WHEN order\_time >= "12:00:00" AND order\_time < "16:00:00" THEN 'Afternoon'

WHEN order\_time >= "16:00:00" AND order\_time < "19:00:00" THEN 'Evening'

WHEN order\_time >= "4:00:00" AND order\_time < "6:00:00" THEN 'Dawn'

ELSE

'Night'

END

AS day\_time\_category

FROM (

SELECT

\*,

EXTRACT(time

FROM

order\_purchase\_timestamp) AS order\_time

FROM

`scalers-sql.Retail\_company.orders`) AS x

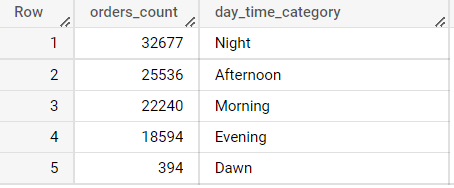
GROUP BY

day\_time\_category

ORDER BY

orders\_count DESC

*Results*



The results imply that *max.* number of orders are getting placed at night, followed by afternoon. *Least* number of orders get placed at dawn.

* 1. At what days of the month more orders take place:

Query

SELECT

order\_day,

COUNT(order\_id) AS orders\_count

FROM (

SELECT

order\_id,

EXTRACT(day

FROM

order\_purchase\_timestamp) AS order\_day

FROM

`scalers-sql.Retail\_company.orders`) AS x

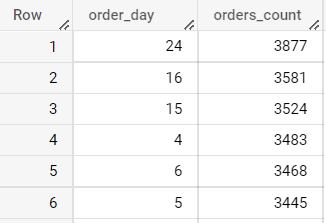
GROUP BY

order\_day

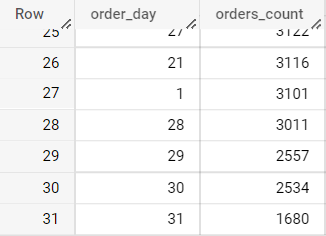
ORDER BY

orders\_count DESC

Result



Most of the orders take place in the middle of the month. And the least number of orders are expected in the last week of the month.



1. **Evolution of e-com orders in Brazil region**
   1. Get month on month orders by region, states,

*For Region/city*

Query

SELECT

COUNT(x.order\_id) AS total\_orders,

x.city,

x.order\_month

FROM (

SELECT

o.order\_id,

c.customer\_city AS city,

c.customer\_state AS state,

EXTRACT(year

FROM

order\_purchase\_timestamp) AS order\_year,

EXTRACT (month

FROM

order\_purchase\_timestamp) AS order\_month

FROM

`scalers-sql.Retail\_company.orders` AS o

JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id) AS x

WHERE

order\_year = 2016

-- WHERE

-- order\_year = 2017

-- WHERE

-- order\_year = 2018

GROUP BY

x.city,x.order\_month

ORDER BY

total\_orders DESC

*As per the requirement we can apply filters using ‘where’ clause which are commented above.*

* + 1. Year 2016



* + 1. Year 2017



* + 1. Year 2018



*For state*

Query

SELECT

COUNT(x.order\_id) AS total\_orders,

x.state,

x.order\_month

FROM (

SELECT

o.order\_id,

c.customer\_city AS city,

c.customer\_state AS state,

EXTRACT(year

FROM

order\_purchase\_timestamp) AS order\_year,

EXTRACT (month

FROM

order\_purchase\_timestamp) AS order\_month

FROM

`scalers-sql.Retail\_company.orders` AS o

JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id) AS x

WHERE

order\_year = 2016

-- WHERE

-- order\_year = 2017

-- WHERE

-- order\_year = 2018

GROUP BY

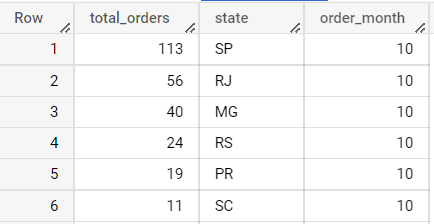
x.state,x.order\_month

ORDER BY

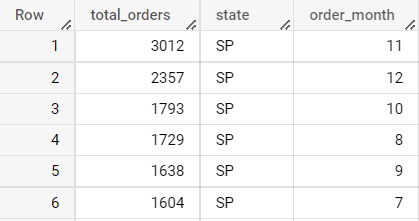
total\_orders DESC

*As per the requirement we can apply filters using ‘where’ clause which are commented above.*

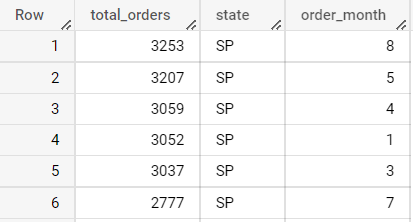
* + 1. Year 2016



* + 1. Year 2017



* + 1. Year 2018



State of São Paulo (SP) has been the top shopper for Retail\_company in the years 2017 and 2018. For the year 2016, due to scarcity of data it would be inappropriate to comment on it. However, the State of São Paulo (SP) being the richest state in Brazil, it is very likely that it would have also been the top shopper in Brazil in 2016 as well.

* 1. How are **customers** distributed in Brazil
     1. As per city,

Query

SELECT

customer\_city,

COUNT(DISTINCT customer\_unique\_id) AS customer\_in\_city

FROM

`scalers-sql.Retail\_company.customers`

GROUP BY

customer\_city

ORDER BY

customer\_in\_city desc

Result



The above mentioned cities are the top 5 cities from which the maximum number of orders come from. ‘Sao Paulo’ (SP) comes at the top and orders the maximum number of products.

* + 1. As per State,

Query

SELECT

customer\_state,

COUNT(DISTINCT customer\_unique\_id) AS customer\_in\_state

FROM

`scalers-sql.Retail\_company.customers`

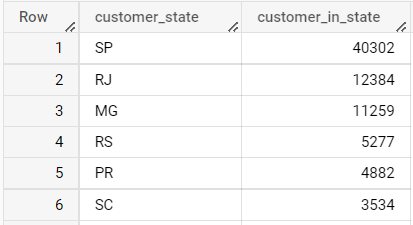
GROUP BY

customer\_state

ORDER BY

customer\_in\_state desc

Result



* 1. How the **number of orders** changing with year in the **cities** of Brazil:

Query

SELECT

COUNT(x.order\_id) AS total\_orders,

x.city

FROM (

SELECT

o.order\_id,

c.customer\_city AS city,

c.customer\_state AS state,

EXTRACT(year

FROM

o.order\_purchase\_timestamp) AS order\_year,

EXTRACT(month

FROM

o.order\_purchase\_timestamp) AS order\_month

FROM

`scalers-sql.Retail\_company.orders` AS o

JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id ) AS x

WHERE

x.order\_year = 2016

-- WHERE

-- x.order\_year = 2017

-- WHERE

-- x.order\_year = 2018

GROUP BY

city

ORDER BY

total\_orders DESC

Result

* + - * 1. Year 2016



* + - * 1. Year 2017



* + - * 1. Year 2018



There is a **17.6%** increase in the product ordering from ‘Retail\_company’ for the city of São Paulo, whereas the growth rates for the cities Rio de Janeiro, Belo Horizonte, and Brasilia are **2.3%**, **12.6%** and **14.16%** respectively.

* 1. How the **number of orders** changing **per** **state** per year:

Query

SELECT

COUNT(x.order\_id) AS total\_orders,

x.state,

FROM (

SELECT

o.order\_id,

c.customer\_city AS city,

c.customer\_state AS state,

EXTRACT(year

FROM

order\_purchase\_timestamp) AS order\_year,

EXTRACT (month

FROM

order\_purchase\_timestamp) AS order\_month

FROM

`scalers-sql.Retail\_company.orders` AS o

JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id) AS x

-- WHERE

-- order\_year = 2016

-- WHERE

-- order\_year = 2017

-- WHERE

-- order\_year = 2018

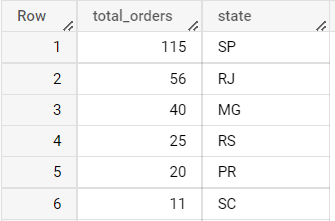
GROUP BY

x.state

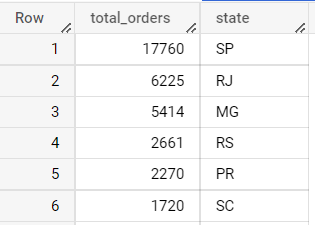
ORDER BY

total\_orders DESC

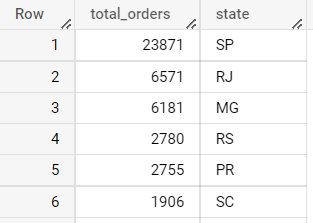
1. Year 2016



1. Year 2017



1. Year 2018

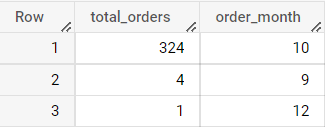


Number of orders being ordered by the customers of the State of São Paulo (SP) increased by **14.67%** from 2017 to 2018. While for the states a very less growth is observed e.g. for the state of Rio de Janeiro, the growth in the number of orders being ordered is only 2.7% only.

* + 1. How the **number of orders** changing **per month** per year:

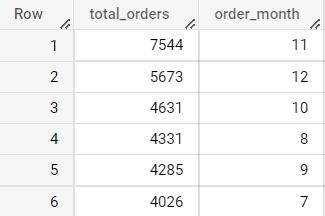
Query

1. Year 2016

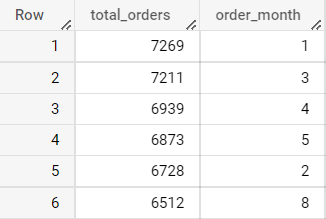


We have only three month purchase data of the customers for this year.

1. Year 2017



1. Year 2018



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

Query

SELECT

limit\_year,

y.sum\_cost\_of\_order,

ROUND((y.sum\_cost\_of\_order - cost\_increase)\*100/(y.sum\_cost\_of\_order + cost\_increase),3) AS perc\_increase

FROM (

SELECT

limit\_year,

ROUND(SUM(cost\_of\_order),3) AS sum\_cost\_of\_order,

LAG(ROUND(SUM(cost\_of\_order),3)) OVER (ORDER BY limit\_year) AS cost\_increase

FROM (

SELECT

\*,

EXTRACT(month

FROM

shipping\_limit\_date) AS limit\_month,

EXTRACT (year

FROM

shipping\_limit\_date) AS limit\_year,

price+ freight\_value AS cost\_of\_order

FROM

`scalers-sql.Retail\_company.order\_items`) AS x

WHERE

(limit\_year = 2017

OR limit\_year = 2018)

AND limit\_month BETWEEN 1

AND 8

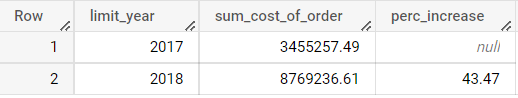
GROUP BY

limit\_year

ORDER BY

limit\_year) AS y

Result



There is a **43.47%** rise in the total cost of orders (price + freight value) for the months between January and August (inclusive) from the year 2017 to 2018.

* 1. Mean and Sum of price and freight value by customer state.

Query

SELECT

x.customer\_state,

ROUND(AVG(price),3) AS Average\_price,

ROUND(SUM(price),3) AS Sum\_price,

ROUND(AVG(freight\_value),3) AS Average\_freight\_value,

ROUND(SUM(freight\_value),3) AS Sum\_freight\_value

FROM (

SELECT

\*

FROM

`scalers-sql.Retail\_company.order\_items` AS ot

JOIN

`scalers-sql.Retail\_company.orders` AS o

ON

ot.order\_id = o.order\_id

JOIN

`scalers-sql.Retail\_company.customers` AS c

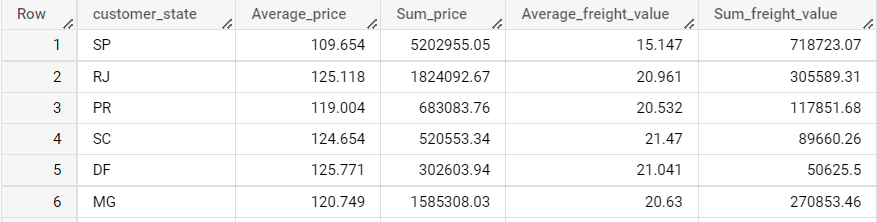
ON

o.customer\_id = c.customer\_id) AS x

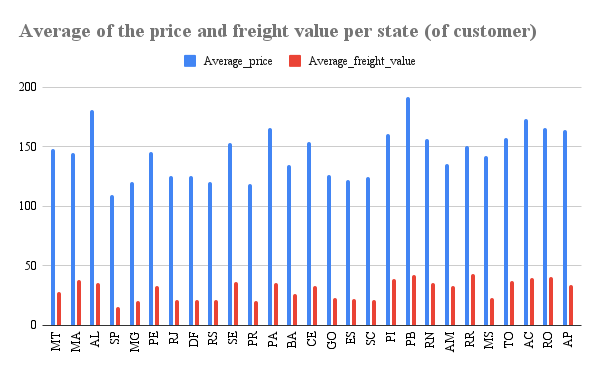
GROUP BY

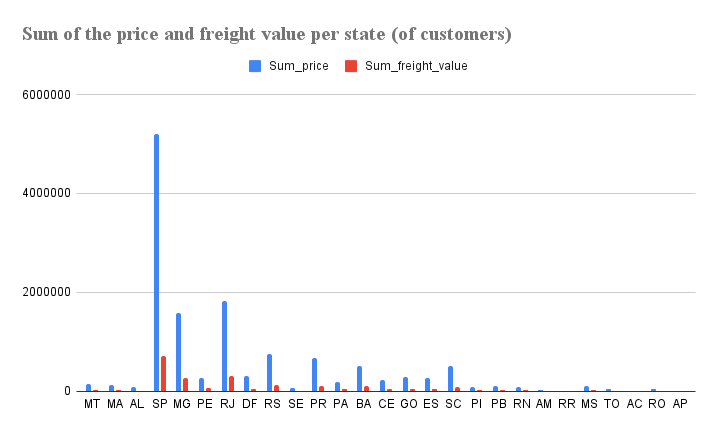
x.customer\_state

Result



A visual representation for each state (of customer) is shown below:





From the plot above observe that SP (i.e. São Paulo) being the richest state in the country, orders most as compared to other states of Brazil from ‘Retail\_company’. This state is important in terms of economy as well as it generates more than **10%** of Brazil's GDP.

Approx **38.28%** of the orders or sale’s total takes place from this state.Next, RJ (Rio de Janeiro), which has around **13.42 %** of total sales followed by MG (Minas Gerais) with **11.66%** of the total sales.

1. Analysis on sales, freight, and others
   1. Calculate days between purchasing, delivering, and estimated delivery

i. Days between purchasing and delivering (as actual\_delivery\_time) and purchasing and estimated delivering (as estimated\_delivery\_time) :

Query

SELECT

\*

FROM (

SELECT

o.order\_id, DATETIME\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day) AS actual\_delivery\_time, DATETIME\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp,day) AS estimated\_delivery\_time

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

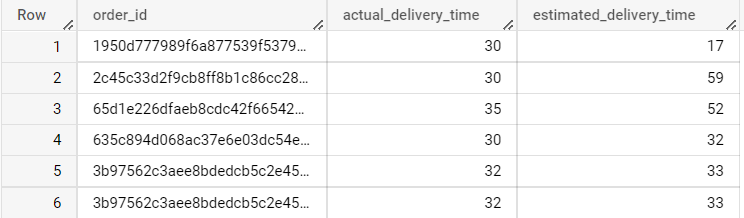
c.customer\_id = o.customer\_id) AS x

WHERE

actual\_delivery\_time IS NOT NULL

AND estimated\_delivery\_time IS NOT NULL

Result



ii. How the estimated delivery and actual delivery time changed over the years:

Query

SELECT

year\_order AS year,

ROUND(AVG(x.actual\_delivery\_time),3) AS avg\_actual\_del\_time,

ROUND(AVG(x.estimated\_delivery\_time),3) AS avg\_estimated\_del\_time,

FROM (

SELECT

o.order\_id,

DATETIME\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day) AS actual\_delivery\_time,

DATETIME\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp,day) AS estimated\_delivery\_time,

EXTRACT(year

FROM

order\_purchase\_timestamp) AS year\_order

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id) AS x

WHERE

actual\_delivery\_time IS NOT NULL

AND estimated\_delivery\_time IS NOT NULL

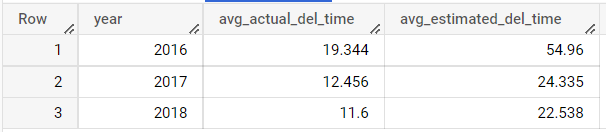
GROUP BY

year

ORDER BY

year

Result



The above data shows that actual delivery time (in days) is decreasing with years and is approximately half of the estimated duration (also mentioned in days).

* 1. Create columns:
     1. Time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date
     2. diff\_estimated\_delivery =

order\_estimated\_delievery\_date - order\_delivered\_customer\_date

Query

SELECT

o.order\_id,

DATETIME\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day) AS time\_to\_delivery, DATETIME\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day) AS diff\_estimated\_delivery

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

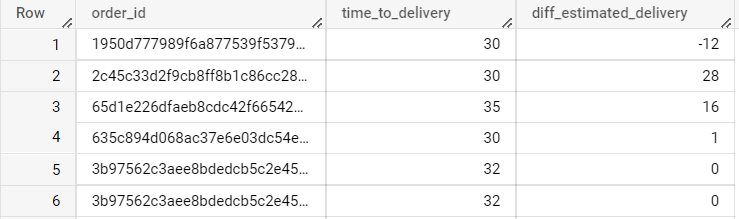
LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

Result



* + 1. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

Query

SELECT

c.customer\_state,

ROUND(AVG(oi.freight\_value),3) AS avg\_freight\_value,

ROUND(AVG(DATETIME\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day)),3) AS avg\_time\_to\_delivery,

ROUND(AVG(DATETIME\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day)),3) AS avg\_diff\_estimated\_delivery

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

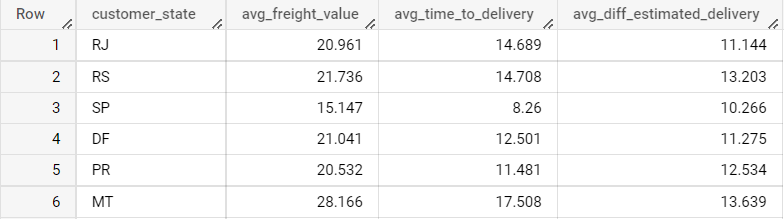
ON

c.customer\_id = o.customer\_id

GROUP BY

c.customer\_state

Result



* + 1. Sort the data to get the following:

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

1. With highest average freight value:

Query

SELECT

c.customer\_state,

ROUND(AVG(oi.freight\_value),3) AS avg\_freight\_value,

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

GROUP BY

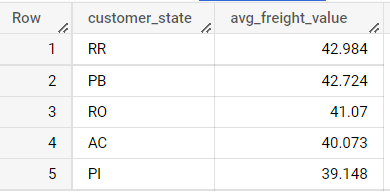
c.customer\_state

ORDER BY

avg\_freight\_value desc

LIMIT 5

Result



1. With lowest average average freight value:

Query

SELECT

c.customer\_state,

ROUND(AVG(oi.freight\_value),3) AS avg\_freight\_value,

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

GROUP BY

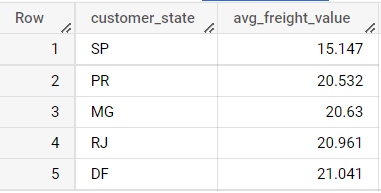
c.customer\_state

ORDER BY

avg\_freight\_value

LIMIT 5

Result



This implies that the richer states have less (almost half )avg\_freight\_value as compared to that of financially weak states in Brazil.

2. **Top 5 states** with highest/lowest average time to delivery

1. With highest average time to delivery

Query

SELECT

c.customer\_state, ROUND(AVG(DATETIME\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day)),3) AS avg\_time\_to\_delivery

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

GROUP BY

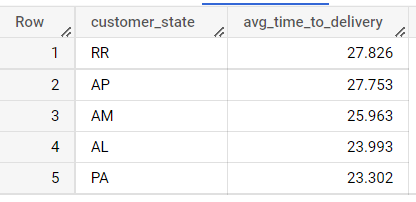
c.customer\_state

ORDER BY

avg\_time\_to\_delivery desc

LIMIT 5

Result



1. With lowest average time to delivery

Query

SELECT

c.customer\_state,

ROUND(AVG(DATETIME\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day)),3) AS avg\_time\_to\_delivery

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

GROUP BY

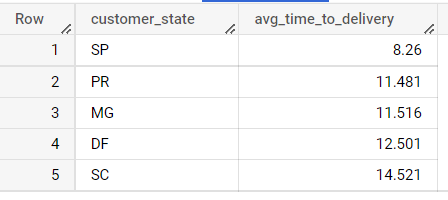
c.customer\_state

ORDER BY

avg\_time\_to\_delivery

LIMIT 5

Result



3. Top 5 states where delivery is really fast/not so fast compared to estimated date

1. With ‘not so fast’ delivery compared to estimated date

Query

SELECT

c.customer\_state, ROUND(AVG(DATETIME\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day)),3) AS avg\_diff\_estimated\_delivery

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

GROUP BY

c.customer\_state

ORDER BY

avg\_diff\_estimated\_delivery

LIMIT 5

Result



1. With ‘really fast’ delivery compared to estimated date

Query

SELECT

c.customer\_state,

ROUND(AVG(DATETIME\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day)),3) AS avg\_diff\_estimated\_delivery

FROM

`scalers-sql.Retail\_company.orders` AS o

LEFT JOIN

`scalers-sql.Retail\_company.order\_items` AS oi

ON

o.order\_id = oi.order\_id

LEFT JOIN

`scalers-sql.Retail\_company.customers` AS c

ON

c.customer\_id = o.customer\_id

GROUP BY

c.customer\_state

ORDER BY

avg\_diff\_estimated\_delivery desc

LIMIT 5

Result



1. **Payment type analysis**
   1. Month over month count of orders for different payment type

Query

SELECT

payment\_type,

month,

orders\_count,

ROUND(((orders\_count - lag\_order\_count)/lag\_order\_count) \*100,3) AS month\_over\_month\_growth -- IN %

FROM (

SELECT

\*,

LAG(orders\_count) OVER (PARTITION BY payment\_type ORDER BY month) AS lag\_order\_count

FROM (

SELECT

COUNT(o.order\_id) AS orders\_count,

p.payment\_type,

EXTRACT(month

FROM

o.order\_purchase\_timestamp) AS month

FROM

`scalers-sql.Retail\_company.orders` AS o

JOIN

`scalers-sql.Retail\_company.payments` AS p

ON

o.order\_id = p.order\_id

GROUP BY

p.payment\_type,

month

ORDER BY

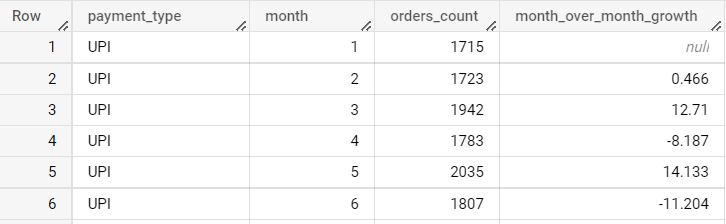
p.payment\_type,

month) AS x ) AS y

ORDER BY

payment\_type

Result



This table & query will show the month over month count of orders as well as month over month growth (in %) for each payment type.

i. Distribution of payments and count of orders:

Query

SELECT

payment\_type,

COUNT(DISTINCT order\_id) AS total\_orders,

ROUND(COUNT(DISTINCT order\_id)\*100/101686,3) AS perc\_total\_orders

FROM

`scalers-sql.Retail\_company.payments`

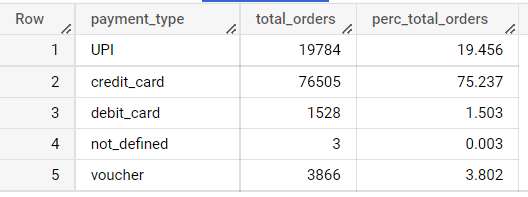
GROUP BY

payment\_type

ORDER BY

payment\_type

Result



As per the data shown above most of the payments take place with credit cards and least number of payments are done with vouchers. 0.003% belongs to an unknown or not defined category.

* 1. Distribution of payment installments and count of orders

There are 25 distinct types of payment installments ranging from 0 to 24 (both inclusive).

Query

SELECT

payment\_installments,

COUNT(DISTINCT order\_id) AS order\_count

FROM

`scalers-sql.Retail\_company.payments`

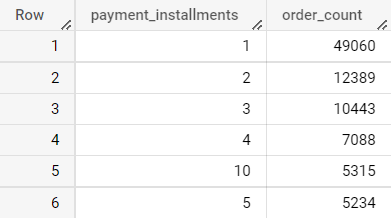
GROUP BY

payment\_installments

ORDER BY

order\_count DESC

Result



As per the above data, we can say that the maximum number of payments are done with one installments only. While the least number of payments are done in 22,23 or 0 installments.

1. **Actionable Insights**
   1. There are **32,951** *unique products*, **73** *product categories*, **3,095** *sellers* in **611** *cities*, in **23** *states* of Brazil. The Company can increase the number of products and categories.
   2. In terms of state, the majority of the *customers* come from the State of São Paulo (SP) and State of Rio de Janeiro i.e. around **40,302** and **12,384** respectively.These is huge difference in these two states. For the rest of the states this difference is even more. The same pattern goes for the major cities of these states as well i.e. the city of São Paulo (**14,984** *customers*) and Rio de Janeiro (**6,620** *customers*) respectively. This indicates that the company needs to work on getting consumers from other *cities,states* as well.
   3. Number of orders from the state of São Paulo increased by **14.67%** from 2017 to 2018. This rise isn’t significant in other states. For instance in the state of Rio de Janeiro, only **2.7%** *rise* was observed for these years. This again proves the point that states other than São Paulo are not performing well enough in terms of sales or purchases from the company. If we take only months between Jan and Aug (both inclusive) the total number of orders for the company increased by **43.47%** from 2017 to 2018. This further indicates that there is growth in the orders with passing years.
   4. The average freight value and average price of the product order by the consumers is *least* i.e. **15.147 Reais** and **109.654 Reais** respectively. While there are some other states where the same is quite high. E.g. For the states RR, PB the average freight value is around **42 Reais**.

*NOTE: Assuming the currency with which the price of the product is described is “Reais”.*

* 1. The state of São Paulo (SP), state of Rio de Janeiro (RJ) and Minas Gerais (MG) together contribute **63.36%** of the total orders for the company. State of São Paulo (SP) contribution is highest i.e. **38.28%**.

The contribution of other states needs to be increased to increase sales.

* 1. Similar to the pattern observed in average freight value above, the pattern can be observed in case of delivery time as well. The average delivery time to deliver a product is the least in the state of São Paulo (**8.26** *days*). While for other states PR, MG and DF average delivery times are **11.481** *days*, **11.516** *days*, **12.501** *days* respectively.
  2. The *fastest deliveries* are made in the state of AC (**20.011** *days*), RO(**19.081** *days*), AM(**18.975** *days*) i.e. where the *difference* between estimated delivery time and actual delivery time is largest. This difference needs to widen up for quicker deliveries.
  3. Majority of the payments are done using credit cards (**75.237%**), followed by UPI service and vouchers with **19.456%** and **3.802%** respectively. In addition, most of the payments for the products are done in 1 installments followed by 2 and 3 number of installments.Least number of payments are done in 22,23, and 0 number of installments. This will require a person from the domain to know which type of payment is more profitable for the company and then take action accordingly.

1. **Recommendations**
   1. Increasing the number of products can help to increase the customer reach. Also, the company can work on how to get sellers from the remaining 4 states as well. Getting more sellers onboard will help the company to *diversify its product range and price*.
   2. In order to get the customers from the other states as well, it is recommended to work on marketing strategies to attract customers and sellers.
   3. The average freight value and average price per product is very less for the state of São Paulo as the maximum number of orders are getting placed from this state only. The average freight value needs to be less for other states as well as it can massively affect the decision of buying the product or not by a consumer. It is recommended to cut down on the average freight value per product per customer especially from states other than SP, and RJ.

*NOTE: Assuming the currency with which the price of the product is described is “Reais”.*

* 1. The state of São Paulo makes **38.28%** of the total orders that the company gets to fulfill. This means that they are the primary shoppers for the company. And since this state is the richest of all in Brazil, this state can be retail\_companyed to increase profit margin, for e.g. Consumers of this region can be exposed to more luxury products.
  2. Similar to freight value, it is recommended to take measures to reduce the average delivery time for consumer satisfaction.
  3. Since a major chunk of consumers are using credit cards for payment, companies can *provide some offers from time to time* on using these credit cards to shop. The same can be done for the consumers who like to pay via UPI. This will motivate the customers to shop more with the company.

Note: *Given all the recommendations above, it should be noted that the conclusions made were with very little data, i.e. only two years of data was given 2017 and 2018 (that was for all the 12 months). Hence the conclusion drawn may not be highly accurate. To increase the confidence more data points must be added.*