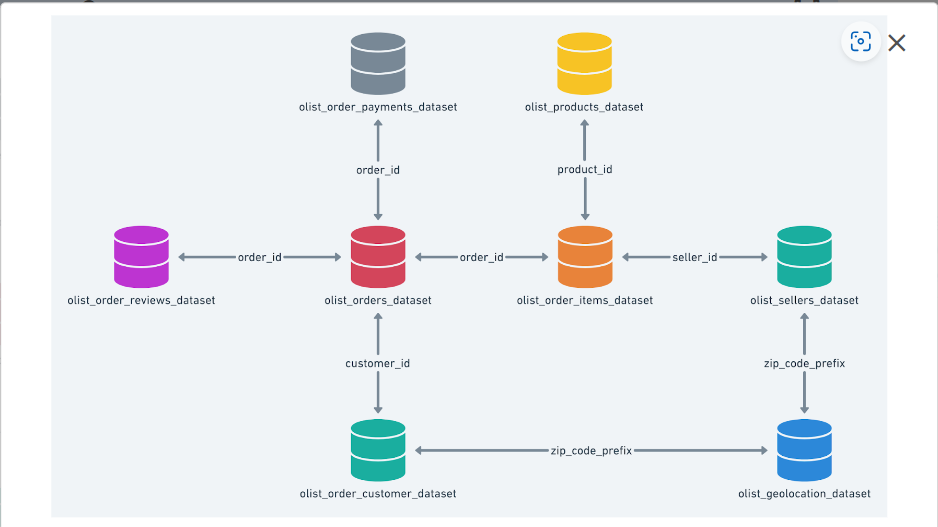
* This business case focuses on the operations of a Retail Company in Brazil.
* The dataset contains 2/3 years of historical data of some orders.
* We need to analyse the dataset to get insights of the operations performed by the retailer.



**Problem Statement:**

Assuming you are a data analyst/ scientist at a Retail Company, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

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.
   2. Get the time range between which the orders were placed.
   3. Count the Cities & States of customers who ordered during the given period.
2. **In-depth Exploration:**  
   1. Is there a growing trend in the no. of orders placed over the past years?
   2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
   3. 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
3. **Evolution of E-commerce orders in the Brazil region:**
   1. Get the month on month no. of orders placed in each state.
   2. How are the customers distributed across all the states?
4. **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).  
      You can use the "payment\_value" column in the payments table to get the cost of orders.
   2. Calculate the Total & Average value of order price for each state.
   3. Calculate the Total & Average value of order freight for each state.
5. **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
   2. Find out the top 5 states with the highest & lowest average freight value.
   3. Find out the top 5 states with the highest & lowest average delivery time.
   4. 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.
6. **Analysis based on the payments:**
   1. Find the month on month no. of orders placed using different payment types.
   2. Find the no. of orders placed on the basis of the payment installments that have been paid.

# Business Case – Target SQL

Q. Data type of all columns in the "customers" table.

  SELECT

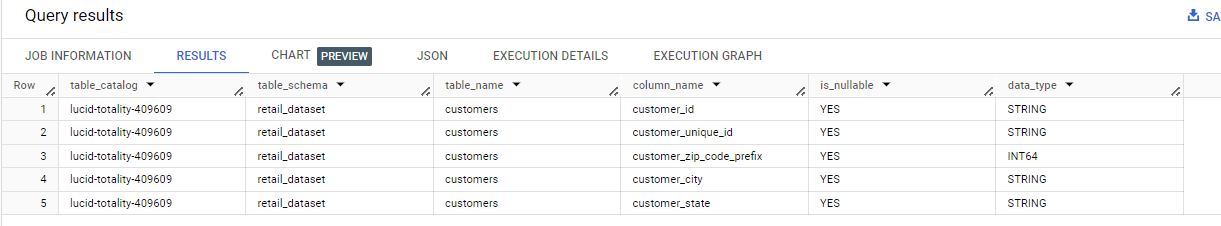
    \* EXCEPT (ordinal\_position, is\_generated, generation\_expression, is\_stored, is\_hidden, is\_updatable, is\_system\_defined, is\_partitioning\_column, clustering\_ordinal\_position, collation\_name, column\_default, rounding\_mode)

  FROM

    `lucid-totality-409609.retail\_dataset`.INFORMATION\_SCHEMA.COLUMNS

  WHERE

    table\_name = 'customers';



Insights:

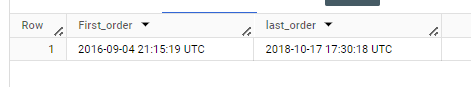
* We could see the customers dataset has 4 columns of String Datatype and 1 column of numerical datatype.
* There are 99441 records in customers table.
* We have 99441 unique and non-null values in customer\_id column.
* We have 96096 unique and non-null values in customer\_unique\_id column.
* There are 14994 unique groups and non-null values in customer\_zip\_code\_prefix column.
* There are 4119 unique groups and non-null values in customer\_city column.
* There are 27 unique groups and non-null values in customer\_state column.

Recommendation: N/A

Q. Get the time range between which the orders were placed.

SELECT MIN(order\_purchase\_timestamp) AS `First\_order`, MAX(order\_purchase\_timestamp) `last\_order`

FROM lucid-totality-409609.retail\_dataset.orders



Insights:

* The orders range between 2016-09-04 21:15:19 UTC AND 2018-10-17 17:30:18 UTC.

Recommendation: N/A

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

SELECT c.customer\_city, COUNT(c.customer\_city) AS `Count\_on\_Customer\_city`

FROM lucid-totality-409609.retail\_dataset.customers AS `c`

JOIN lucid-totality-409609.retail\_dataset.orders AS `o`

  ON o.customer\_id = c.customer\_id

WHERE o.order\_purchase\_timestamp BETWEEN

  (SELECT MIN(order\_purchase\_timestamp) FROM lucid-totality-409609.retail\_dataset.orders) AND

  (SELECT MAX(order\_purchase\_timestamp) FROM lucid-totality-409609.retail\_dataset.orders)

  GROUP BY c.customer\_city



SELECT c.customer\_state, COUNT(c.customer\_state) AS `Count\_on\_Customer\_state`

FROM lucid-totality-409609.retail\_dataset.customers AS `c`

JOIN lucid-totality-409609.retail\_dataset.orders AS `o`

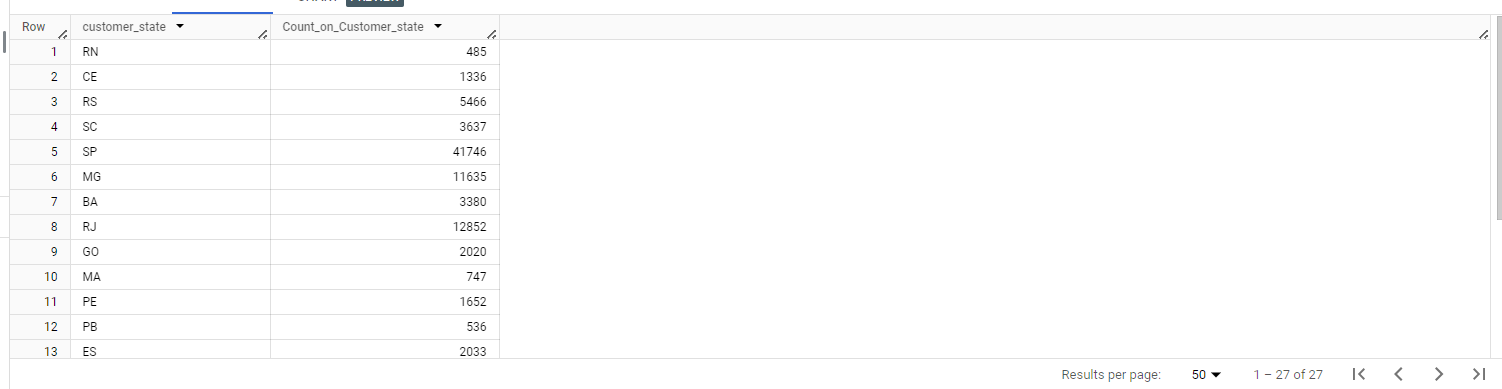
  ON o.customer\_id = c.customer\_id

WHERE o.order\_purchase\_timestamp BETWEEN

  (SELECT MIN(order\_purchase\_timestamp) FROM lucid-totality-409609.retail\_dataset.orders) AND

  (SELECT MAX(order\_purchase\_timestamp) FROM lucid-totality-409609.retail\_dataset.orders)

  GROUP BY c.customer\_state



Insights:

* The customers are from 4119 different cities from 27 different states.

Recommendation: N/A

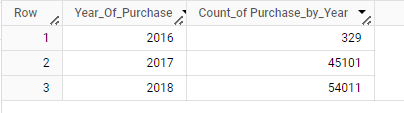
Q. Is there a growing trend in the no. of orders placed over the past years?

SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year\_Of\_Purchase`,

  COUNT(order\_purchase\_timestamp) OVER(PARTITION BY EXTRACT(YEAR FROM order\_purchase\_timestamp)) AS `Count\_of Purchase\_by\_Year`

FROM `lucid-totality-409609.retail\_dataset.orders`

ORDER BY Year\_Of\_Purchase ASC



Insights:

* Compared to 2016, the numbers of 2017 had a huge growth.
* Compared to 2017, the count of 2018 is higher
* We could see a growing trend from later years

Recommendation: N/A

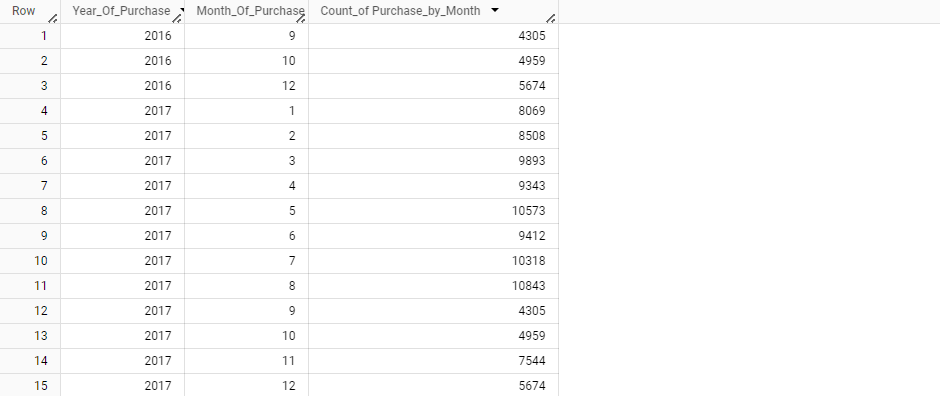
SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year\_Of\_Purchase`,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month\_Of\_Purchase`,

  COUNT(order\_purchase\_timestamp) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp)) AS `Count\_of Purchase\_by\_Month`

FROM `lucid-totality-409609.retail\_dataset.orders`

ORDER BY Year\_Of\_Purchase ASC, Month\_Of\_Purchase ASC



Insights:

* In 2016, We could see upward trend over the months of October, November and December
* In 2017, There was a steady growth and only during September and December, there was a slight decrease in the orders.
* In 2018, The orders were consistently rising and dropping between 6167 and 7269 till August and the orders were not good during September and October as there was very less purchase.

Recommendation: N/A

Q. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

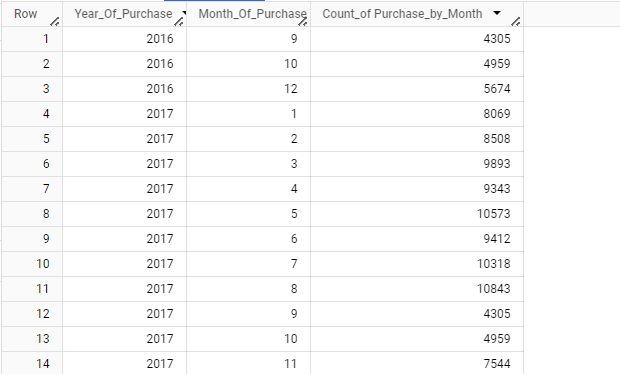
SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year\_Of\_Purchase`,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month\_Of\_Purchase`,

  COUNT(order\_purchase\_timestamp) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp)) AS `Count\_of Purchase\_by\_Month`

FROM `lucid-totality-409609.retail\_dataset.orders`

ORDER BY Year\_Of\_Purchase ASC, Month\_Of\_Purchase ASC



Insights:

* We can only find a pattern during December month among the years 2016,2017,2018 as there is a drop in the number of orders.

Recommendation: N/A

Q. 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 DISTINCT T.TimeOfDay, COUNT(\*) OVER(PARTITION BY T.TimeOfDay) AS `Within\_Period\_Count\_of\_Orders`

FROM (

  SELECT order\_purchase\_timestamp,

  (

    CASE

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

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

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

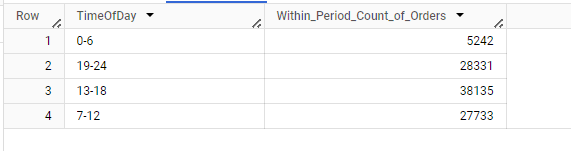
    ELSE '19-24'

    END

  ) AS `TimeOfDay`

FROM `lucid-totality-409609.retail\_dataset.orders`

) AS `T`



Insights:

* We can see the most number of orders were placed between 13 and 18 hours which is during Afternoon time.
* There are purchases made all the time but comparatively only during Dawn we have very less when we consider the period between 2016 – 2018.

Recommendation: N/A

Q. Get the month on month no. of orders placed in each state.

/\*Get the month on month no. of orders placed in each state.\*/

SELECT DISTINCT c.customer\_state, EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS `MonthOfPurchase`,

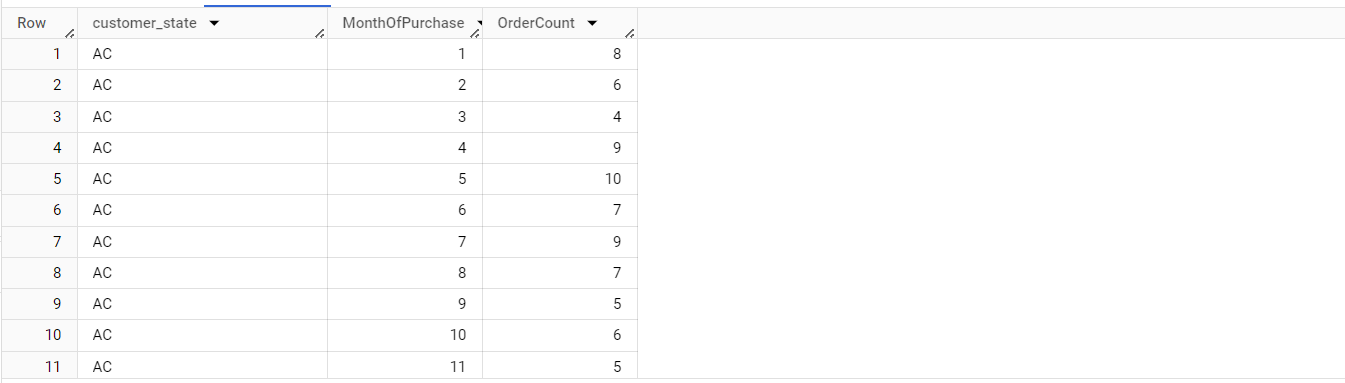
  COUNT(o.order\_id) OVER(PARTITION BY c.customer\_state, EXTRACT(MONTH FROM o.order\_purchase\_timestamp)) AS `OrderCount`

FROM `lucid-totality-409609.retail\_dataset.customers` AS `c`

JOIN `lucid-totality-409609.retail\_dataset.orders` AS `o`

  ON o.customer\_id = c.customer\_id

ORDER BY c.customer\_state ASC, MonthOfPurchase ASC



Insights:

* N/A

Recommendation: N/A

Q. How are the customers distributed across all the states?

  /\*How are the customers distributed across all the states?\*/

SELECT

  customer\_state,

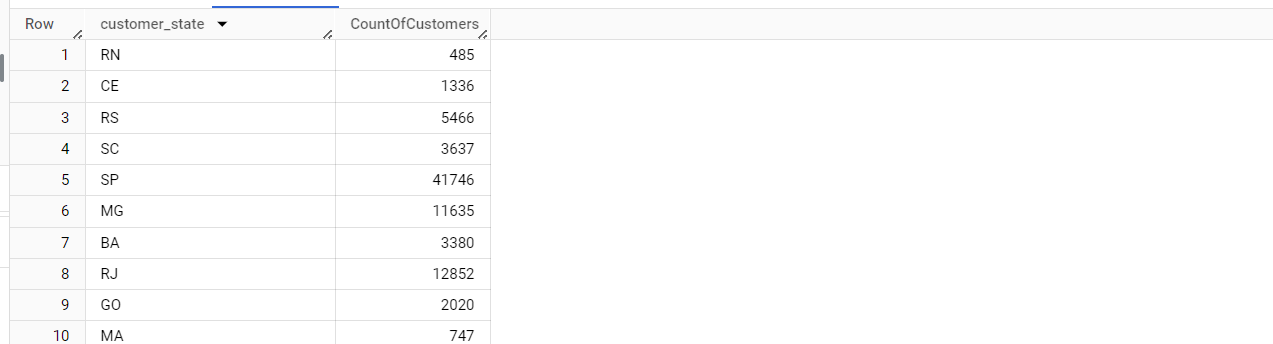
  COUNT(\*) AS `CountOfCustomers`

FROM

  `lucid-totality-409609.retail\_dataset.customers`

GROUP BY

  customer\_state



/\*Below table displays the result by descending order of count of customer\*/

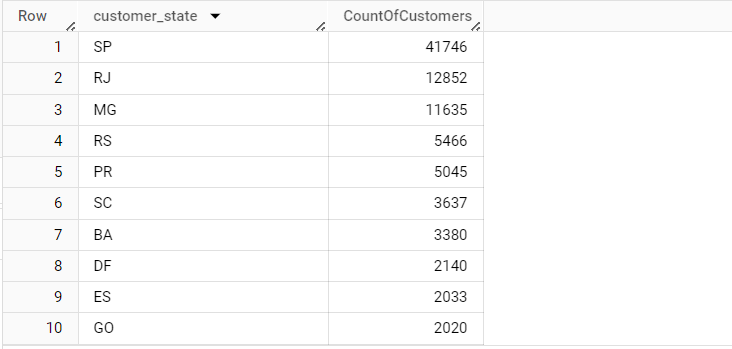
SELECT DISTINCT customer\_state, COUNT(customer\_unique\_id) OVER(PARTITION BY customer\_state) AS `Number\_Of\_Customers\_Per\_state`,

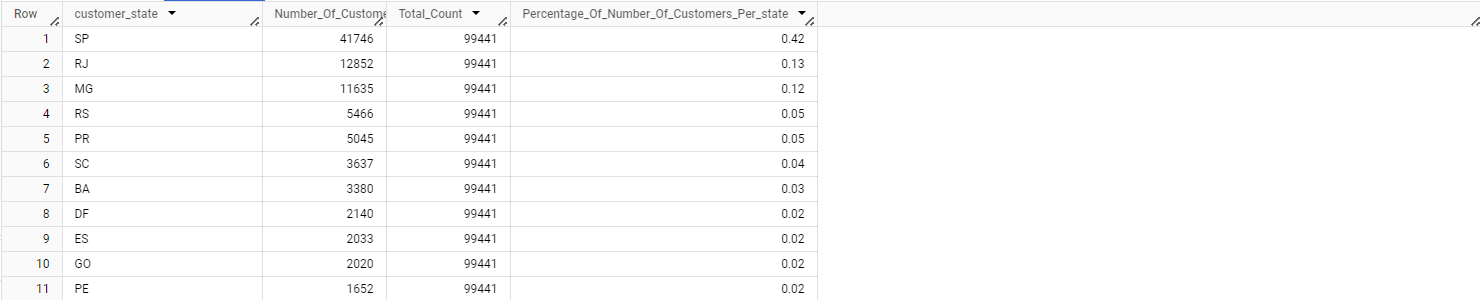
  COUNT(customer\_unique\_id) OVER() AS `Total\_Count`,

  ROUND(COUNT(customer\_unique\_id) OVER(PARTITION BY customer\_state) / COUNT(customer\_unique\_id) OVER(),2) AS `Percentage\_Of\_Number\_Of\_Customers\_Per\_state`,

FROM `lucid-totality-409609.retail\_dataset.customers`

ORDER BY Number\_Of\_Customers\_Per\_state DESC





Insights:

* We can see the top 5 states where the customers are in majority are SP, RJ, MG, RS, PR.

Recommendation: N/A

Q. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(YEAR FROM order\_purchase\_timestamp)) AS `Total\_Purchase\_Value\_Based\_on\_Years`

FROM `lucid-totality-409609.retail\_dataset.orders` AS `o`

JOIN `lucid-totality-409609.retail\_dataset.payments` AS `p`

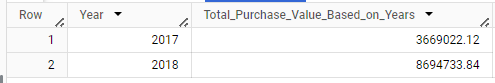
  ON p.order\_id = o.order\_id

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) != 2016

  AND

  EXTRACT(MONTH FROM order\_purchase\_timestamp) NOT IN (9,10,11,12)

ORDER BY Year ASC



SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(YEAR FROM order\_purchase\_timestamp)) AS `Total\_Purchase\_Value\_Based\_on\_Years`,

  SUM(payment\_value) OVER() AS `Total\_value`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(YEAR FROM order\_purchase\_timestamp))/ SUM(payment\_value) OVER() AS `Percentage\_of\_Payments\_By\_Years`

FROM `lucid-totality-409609.retail\_dataset.orders` AS `o`

JOIN `lucid-totality-409609.retail\_dataset.payments` AS `p`

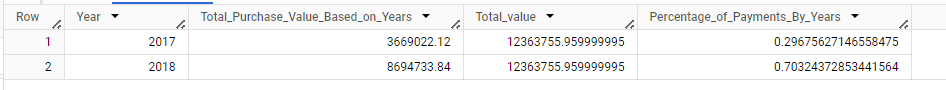
  ON p.order\_id = o.order\_id

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) != 2016

  AND

  EXTRACT(MONTH FROM order\_purchase\_timestamp) NOT IN (9,10,11,12)

ORDER BY Year ASC#, Month ASC



SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp)) AS `Total\_Purchase\_Value\_Based\_on\_Months`,

  SUM(payment\_value) OVER() AS `Total\_value`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp))/ SUM(payment\_value) OVER() AS `Percentage\_of\_Payments\_By\_Months\_2017`

FROM `lucid-totality-409609.retail\_dataset.orders` AS `o`

JOIN `lucid-totality-409609.retail\_dataset.payments` AS `p`

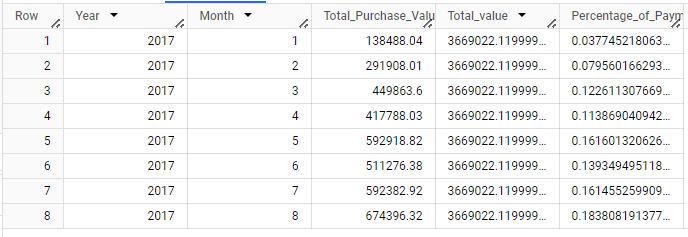
  ON p.order\_id = o.order\_id

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) NOT IN (2016, 2018)

  AND

  EXTRACT(MONTH FROM order\_purchase\_timestamp) NOT IN (9,10,11,12)

ORDER BY Year ASC, Month ASC



SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp)) AS `Total\_Purchase\_Value\_Based\_on\_Months`,

  SUM(payment\_value) OVER() AS `Total\_value`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp))/ SUM(payment\_value) OVER() AS `Percentage\_of\_Payments\_By\_Months\_2018`

FROM `lucid-totality-409609.retail\_dataset.orders` AS `o`

JOIN `lucid-totality-409609.retail\_dataset.payments` AS `p`

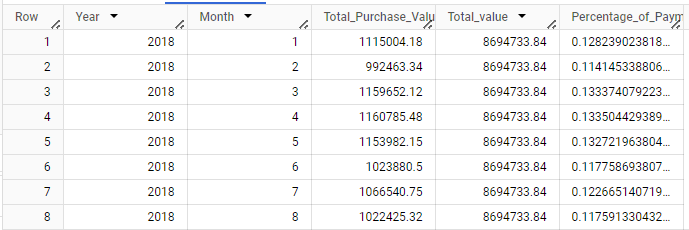
  ON p.order\_id = o.order\_id

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) NOT IN (2016, 2017)

  AND

  EXTRACT(MONTH FROM order\_purchase\_timestamp) NOT IN (9,10,11,12)

ORDER BY Year ASC, Month ASC



WITH `2018` AS

(

  SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp)) AS `Total\_Purchase\_Value\_Based\_on\_Months`,

  SUM(payment\_value) OVER() AS `Total\_value`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp))/ SUM(payment\_value) OVER() AS `Percentage\_of\_Payments\_By\_Months\_2018`

FROM `lucid-totality-409609.retail\_dataset.orders` AS `o`

JOIN `lucid-totality-409609.retail\_dataset.payments` AS `p`

  ON p.order\_id = o.order\_id

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) NOT IN (2016, 2017)

  AND

  EXTRACT(MONTH FROM order\_purchase\_timestamp) NOT IN (9,10,11,12)

ORDER BY Year ASC, Month ASC

),

`2017` AS

(

  SELECT DISTINCT EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp)) AS `Total\_Purchase\_Value\_Based\_on\_Months`,

  SUM(payment\_value) OVER() AS `Total\_value`,

  SUM(payment\_value) OVER(PARTITION BY EXTRACT(MONTH FROM order\_purchase\_timestamp))/ SUM(payment\_value) OVER() AS `Percentage\_of\_Payments\_By\_Months\_2017`

FROM `lucid-totality-409609.retail\_dataset.orders` AS `o`

JOIN `lucid-totality-409609.retail\_dataset.payments` AS `p`

  ON p.order\_id = o.order\_id

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) NOT IN (2016, 2018)

  AND

  EXTRACT(MONTH FROM order\_purchase\_timestamp) NOT IN (9,10,11,12)

ORDER BY Year ASC, Month ASC

)

SELECT T17.Year, T17.Month, T17.Percentage\_of\_Payments\_By\_Months\_2017, T18.Year, T18.Month, T18.Percentage\_of\_Payments\_By\_Months\_2018,

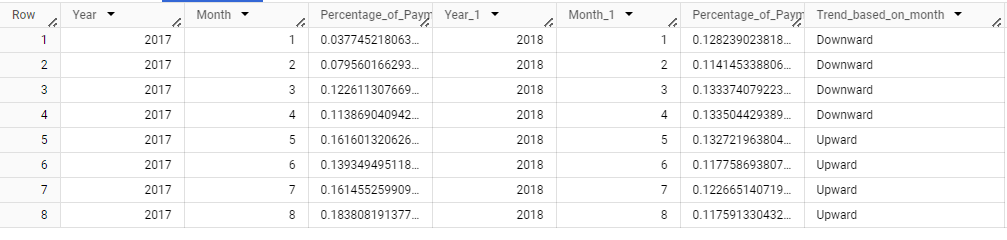
  IF(T17.Percentage\_of\_Payments\_By\_Months\_2017 < T18.Percentage\_of\_Payments\_By\_Months\_2018, 'Downward', 'Upward') AS `Trend\_based\_on\_month`

FROM `2017` AS `T17`

JOIN `2018` AS `T18`

  ON T18.Month = T17.Month

ORDER BY T17.Month ASC



Insights:

* Comparing 2017 and 2018, we could see for the first 4 months, 2017 had less purchase and 2018 had higher purchase.
* Similarly Comparing 2017 and 2018, we could see for the last 4 months, 2017 had high purchase and 2018 had lesser purchase.

Recommendation: N/A

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

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

SELECT DISTINCT c.customer\_state,

  SUM(payment\_value) OVER(PARTITION BY c.customer\_state) AS `Total\_Value\_Based\_On\_State`,

  AVG(payment\_value) OVER(PARTITION BY c.customer\_state) AS `Average\_Value\_Based\_On\_State`

FROM `lucid-totality-409609.retail\_dataset.payments` AS `p`

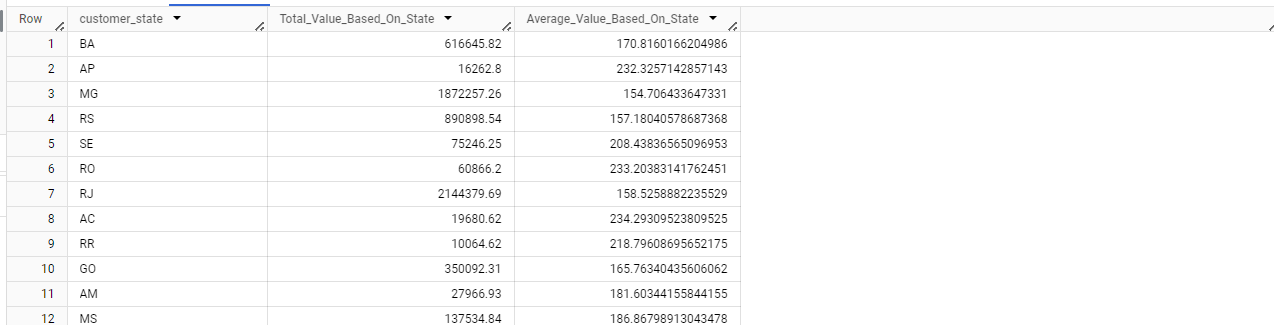
JOIN `lucid-totality-409609.retail\_dataset.orders` AS `o`

  ON o.order\_id = p.order\_id

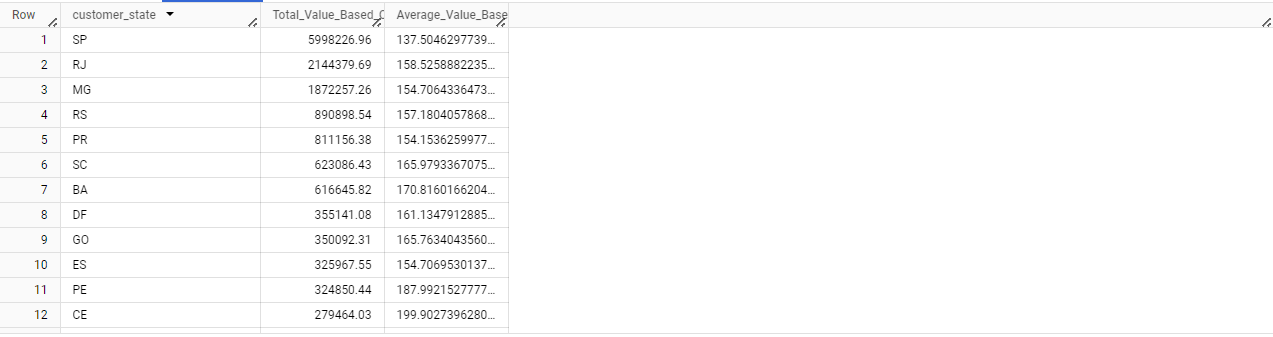
JOIN `lucid-totality-409609.retail\_dataset.customers` AS `c`

  ON c.customer\_id = o.customer\_id

#  ORDER BY `Total\_Value\_Based\_On\_State` DESC, `Average\_Value\_Based\_On\_State` DESC



/\*Sorted Data based on Total Desc\*/



Insights:

* The top 5 states contribute 73% of the total sales.
* The top 5 states are SP, RJ, MG, RS and PR.

Recommendation: N/A

Q. Calculate the Total & Average value of order freight for each state.

SELECT DISTINCT c.customer\_state,

  SUM(oi.freight\_value) OVER(PARTITION BY c.customer\_state) AS `Total\_Freight\_Value\_By\_State`,

  AVG(oi.freight\_value) OVER(PARTITION BY c.customer\_state) AS `Average\_Freight\_Value\_By\_State`

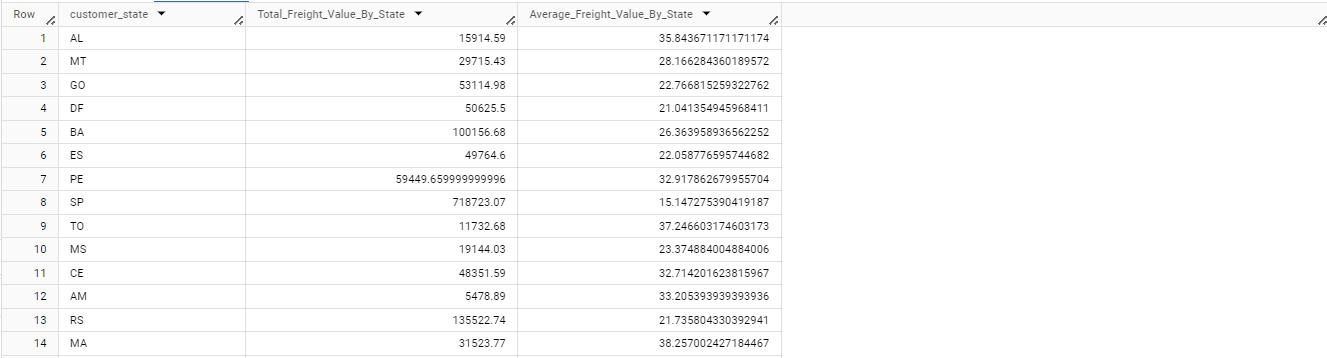
FROM lucid-totality-409609.retail\_dataset.customers AS `c`

JOIN lucid-totality-409609.retail\_dataset.orders AS `o`

  ON o.customer\_id = c.customer\_id

JOIN lucid-totality-409609.retail\_dataset.order\_items AS `oi`

  ON oi.order\_id = o.order\_id



Q. 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

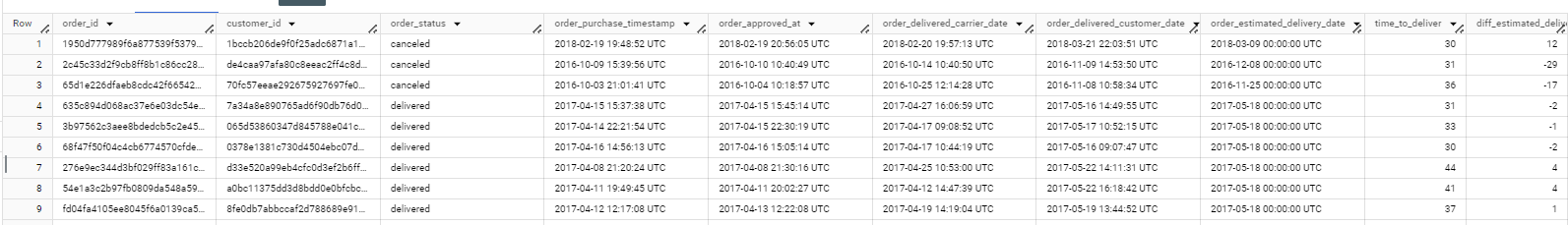
diff\_estimated\_delivery = order\_delivered\_customer\_date - order\_estimated\_delivery\_date

SELECT \*,

    DATETIME\_DIFF (DATETIME(order\_delivered\_customer\_date), DATETIME(order\_purchase\_timestamp), DAY) AS `time\_to\_deliver`,

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

FROM `lucid-totality-409609.retail\_dataset.orders`



Insights:

* We can see there are some positive values for diff\_estimated\_delivery which tells that the delivery was made post the agreed time/day.
* We can see there are some negative values for diff\_estimated\_delivery which tells that the delivery was made within the agreed time/day.

Recommendation: N/A

Q. Find out the top 5 states with the highest & lowest average freight value.

WITH `Last5FreightValues` AS

(

  SELECT DISTINCT c.customer\_state,

  SUM(oi.freight\_value) OVER(PARTITION BY c.customer\_state) AS `Total\_Freight\_Value\_By\_State`,

  AVG(oi.freight\_value) OVER(PARTITION BY c.customer\_state) AS `Average\_Freight\_Value\_By\_State`

FROM lucid-totality-409609.retail\_dataset.customers AS `c`

JOIN lucid-totality-409609.retail\_dataset.orders AS `o`

  ON o.customer\_id = c.customer\_id

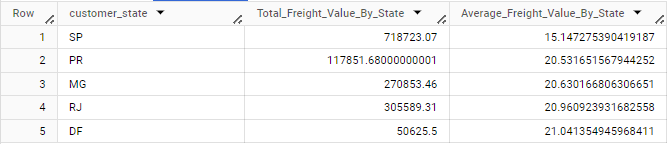
JOIN lucid-totality-409609.retail\_dataset.order\_items AS `oi`

  ON oi.order\_id = o.order\_id

ORDER BY Average\_Freight\_Value\_By\_State ASC

LIMIT 5

),



WITH `Top5FreightValues` AS

(SELECT DISTINCT c.customer\_state,

  SUM(oi.freight\_value) OVER(PARTITION BY c.customer\_state) AS `Total\_Freight\_Value\_By\_State`,

  AVG(oi.freight\_value) OVER(PARTITION BY c.customer\_state) AS `Average\_Freight\_Value\_By\_State`

FROM lucid-totality-409609.retail\_dataset.customers AS `c`

JOIN lucid-totality-409609.retail\_dataset.orders AS `o`

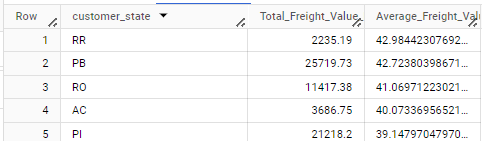
  ON o.customer\_id = c.customer\_id

JOIN lucid-totality-409609.retail\_dataset.order\_items AS `oi`

  ON oi.order\_id = o.order\_id

ORDER BY Average\_Freight\_Value\_By\_State DESC

LIMIT 5)



Insights:

* We can see the top 5 states where the average freight value is high based on customer states are RR,PB,RO,AC,PI.
* We can see the last 5 states where the average freight value is high based on customer states are DF,RJ,MG.PR,SP.

Recommendation: N/A

Q. Find out the top 5 states with the highest & lowest average delivery time.

/\*States with highest delivery time\*/

SELECT

  DISTINCT customer\_state,

  AVG(DATETIME\_DIFF (DATETIME(order\_delivered\_customer\_date), DATETIME(order\_purchase\_timestamp), DAY) ) OVER(PARTITION BY customer\_state) AS `AverageDeliveryTime`

FROM

  `lucid-totality-409609.retail\_dataset.customers` AS `cust`

JOIN

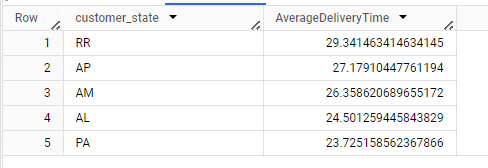
  `lucid-totality-409609.retail\_dataset.orders` AS `ord`

ON

  ord.customer\_id = cust.customer\_id

ORDER BY AverageDeliveryTime DESC

LIMIT 5



SELECT

  DISTINCT customer\_state,

  AVG(DATETIME\_DIFF (DATETIME(order\_delivered\_customer\_date), DATETIME(order\_purchase\_timestamp), DAY) ) OVER(PARTITION BY customer\_state) AS `AverageDeliveryTime`

FROM

  `lucid-totality-409609.retail\_dataset.customers` AS `cust`

JOIN

  `lucid-totality-409609.retail\_dataset.orders` AS `ord`

ON

  ord.customer\_id = cust.customer\_id

ORDER BY AverageDeliveryTime ASC

LIMIT 5

/\*States with lowest delivery time\*/



Insights:

* The 5 states were the delivery time is very high are RR,AP,AM,AL,PA.
* RR state customers get their orders very delayed.
* The 5 states were the delivery time is very low are SC,DF,MG,PR, SP.
* SP state customers get their orders delivered quickly compared to the other states.

Recommendation: N/A

Q. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

  /\*Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.\*/

SELECT

  DISTINCT cust.customer\_state,

  AVG(DATETIME\_DIFF (DATETIME(order\_delivered\_customer\_date), DATETIME(order\_estimated\_delivery\_date), DAY)) OVER(PARTITION BY cust.customer\_state) AS `diff\_estimated\_delivery\_average`

FROM

  `lucid-totality-409609.retail\_dataset.customers` AS `cust`

JOIN

  `lucid-totality-409609.retail\_dataset.orders` AS `ord`

ON

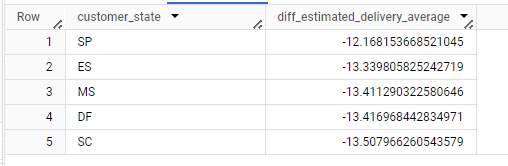
  ord.customer\_id = cust.customer\_id

WHERE

  ord.order\_delivered\_customer\_date < ord.order\_estimated\_delivery\_date

ORDER BY diff\_estimated\_delivery\_average DESC

LIMIT 5



Insights:

* The 5 states were the average delivery time is very low are SC,DF,MG,PR, SP.
* SP state customers get their orders delivered quickly compared to the other states.

Recommendation: N/A

Q. Find the month on month no. of orders placed using different payment types.

/\*Find the month on month no. of orders placed using different payment types.\*/

SELECT T1.\* EXCEPT(NumberOfOrdersBasedOnMonthsandPaymentType),

    T1.NumberOfOrdersBasedOnMonthsandPaymentType - IF(LAG(T1.NumberOfOrdersBasedOnMonthsandPaymentType) OVER(PARTITION BY T1.payment\_type ORDER BY T1.payment\_type,/\*T1.Year,\*/ T1.Month ) IS NULL, 0, LAG(T1.NumberOfOrdersBasedOnMonthsandPaymentType) OVER(PARTITION BY T1.payment\_type ORDER BY T1.payment\_type,/\*T1.Year,\*/ T1.Month )) AS `CountOfOrdersSeparatedByPaymentTypeMonth`

FROM

(

SELECT

  DISTINCT pay.payment\_type,

/\*  EXTRACT(YEAR

  FROM

    ord.order\_purchase\_timestamp) AS `Year`,\*/

  EXTRACT(MONTH

  FROM

    ord.order\_purchase\_timestamp) AS `Month`,

  COUNT(ord.order\_id) OVER(PARTITION BY pay.payment\_type ORDER BY

  /\*EXTRACT(YEAR

  FROM

    ord.order\_purchase\_timestamp),\*/

  EXTRACT(MONTH

  FROM

    ord.order\_purchase\_timestamp)) AS `NumberOfOrdersBasedOnMonthsandPaymentType`

FROM

  `lucid-totality-409609.retail\_dataset.payments` AS `pay`

JOIN

  `lucid-totality-409609.retail\_dataset.orders` AS `ord`

ON

  ord.order\_id = pay.order\_id

JOIN

  `lucid-totality-409609.retail\_dataset.customers` AS `cust`

ON

  ord.customer\_id = cust.customer\_id

ORDER BY

  payment\_type ASC,

  #Year ASC,

  Month ASC

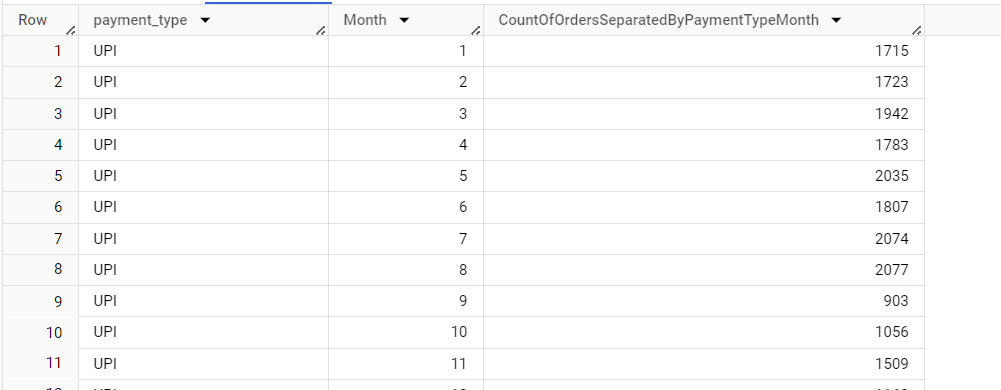
) AS `T1`

ORDER BY

  T1.payment\_type ASC,

  #T1.Year ASC,

  T1.Month ASC



/\*Find the month on month no. of orders placed using different payment types.

Additionally segregated by year\*/

SELECT T1.\* EXCEPT(NumberOfOrdersBasedOnYearMonthsandPaymentType),

    T1.NumberOfOrdersBasedOnYearMonthsandPaymentType - IF(LAG(T1.NumberOfOrdersBasedOnYearMonthsandPaymentType) OVER(PARTITION BY T1.payment\_type ORDER BY T1.payment\_type,T1.Year, T1.Month ) IS NULL, 0, LAG(T1.NumberOfOrdersBasedOnYearMonthsandPaymentType) OVER(PARTITION BY T1.payment\_type ORDER BY T1.payment\_type,T1.Year, T1.Month )) AS `CountOfOrdersSeparatedByPaymentTypeYearMonth`

FROM

(

SELECT

  DISTINCT pay.payment\_type,

  EXTRACT(YEAR

  FROM

    ord.order\_purchase\_timestamp) AS `Year`,

  EXTRACT(MONTH

  FROM

    ord.order\_purchase\_timestamp) AS `Month`,

  COUNT(ord.order\_id) OVER(PARTITION BY pay.payment\_type ORDER BY

  EXTRACT(YEAR

  FROM

    ord.order\_purchase\_timestamp),

  EXTRACT(MONTH

  FROM

    ord.order\_purchase\_timestamp)) AS `NumberOfOrdersBasedOnYearMonthsandPaymentType`

FROM

  `lucid-totality-409609.retail\_dataset.payments` AS `pay`

JOIN

  `lucid-totality-409609.retail\_dataset.orders` AS `ord`

ON

  ord.order\_id = pay.order\_id

JOIN

  `lucid-totality-409609.retail\_dataset.customers` AS `cust`

ON

  ord.customer\_id = cust.customer\_id

ORDER BY

  payment\_type ASC,

  Year ASC,

  Month ASC

) AS `T1`

ORDER BY

  T1.payment\_type ASC,

  T1.Year ASC,

  T1.Month ASC



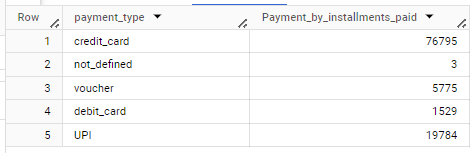
Insights: N/A

Recommendation: N/A

Q. Find the no. of orders placed on the basis of the payment installments that have been paid.

SELECT DISTINCT payment\_type, COUNT(payment\_sequential) OVER(PARTITION BY payment\_type) AS `Payment\_by\_installments\_paid`

FROM `lucid-totality-409609.retail\_dataset.payments`



Insights:

* We could see the orders were mostly paid using credit card.
* Not defined payment type can be assumed as Paid by cash.

Recommendation: N/A