[redacted] SQL Business Case

❖ Topic: SQL

❖ Submitted by Piyush Joshi-DSML-October-2023

I. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

A. Data type of all columns in the “customers” table

```
SELECT *
FROM `ecommerce-406315.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name='customers';
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	table_name	column_name	ordinal_position	is_nullable	data_type	is_generated
1	customers	customer_id	1	YES	STRING	NEVER
2	customers	customer_unique_id	2	YES	STRING	NEVER
3	customers	customer_zip_code_prefix	3	YES	INT64	NEVER
4	customers	customer_city	4	YES	STRING	NEVER
5	customers	customer_state	5	YES	STRING	NEVER

INSIGHT: The above-mentioned query displays the data type of all columns in the “customers” table from ‘ecommerce-406315’ catalog within the "[redacted]" dataset.

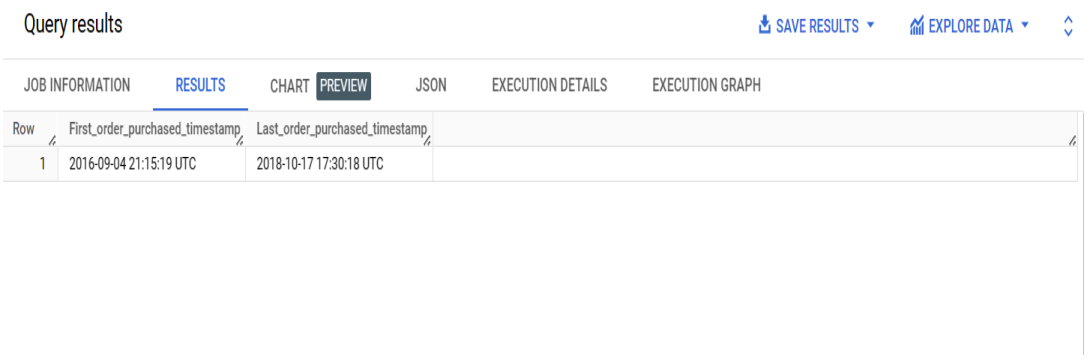
-It shows that all columns in this table are “nullable” which is a default value of a column which can hold null values.

- All columns data type are string except column `customer_zip_code_prefix` which has integer data type.

RECOMMENDATION: To fellow data analyst who would want to remove the null constraint, a NOT NULL constraint can be applied on one of the column with unique values; later, applying a primary key on that same column can be done.

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

```
SELECT
MIN(order_purchase_timestamp)`First_order_purchased_timestamp`,
MAX(order_purchase_timestamp)`Last_order_purchased_timestamp`
FROM `target.orders`;
```

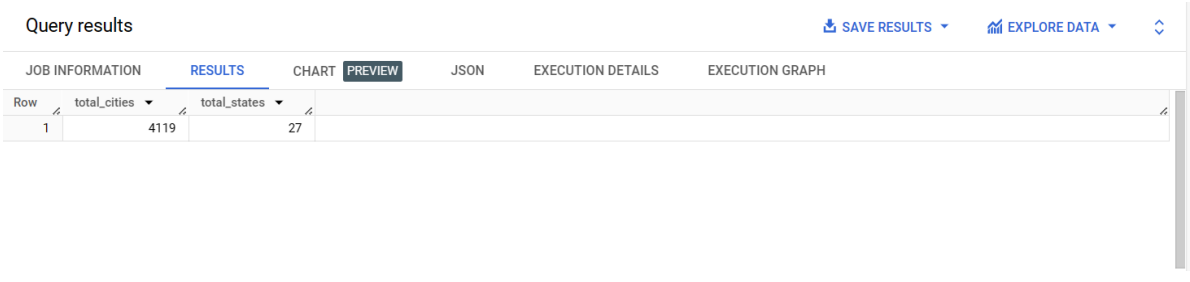


INSIGHT: The result displays the orders were placed between 4 September 2019, 09:15 PM UTC to 17 October 2018, 05:30:18 PM UTC.

RECOMMENDATION: To fellow data analysts, UTC time zones to local time zones can be taken note of so that time of the day and region-specific analysis can be more in-depth and true.

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

```
SELECT COUNT(DISTINCT c.customer_city) AS total_cities,
COUNT(DISTINCT c.customer_state) AS total_states
FROM target.customers as c
INNER JOIN `target.orders` as o
ON c.customer_id=o.customer_id
WHERE order_purchase_timestamp BETWEEN (select min(order_purchase_timestamp)
from `target.orders` ) AND
(select max(order_purchase_timestamp) from `target.orders`);
```



INSIGHT: The result displays the company has a foothold in 4119 cities across 27 cities from the records of the orders placed between 4 September 2019, 09:15 PM UTC to 17 October 2018, 05:30:18 PM UTC.

RECOMMENDATION: The data can further be used to count the outlier cities and states for subsequent sales expansion.

II. In-depth Exploration:

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

```
SELECT T.*,
ROUND(AVG(T.`Number_of_Orders`) OVER (PARTITION BY `Years` ORDER BY
T.`Months` ASC  ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW),2) as
`MOVING_AVG`
FROM
(SELECT EXTRACT(YEAR FROM order_purchase_timestamp)`Years`,
       EXTRACT(MONTH FROM order_purchase_timestamp)`Months`,
       COUNT(order_id) AS `Number_of_Orders`
FROM `target.orders`
GROUP BY `Years`,`Months`
ORDER BY `Years` ASC,`Months` ASC)T
ORDER BY `Years` ASC,`Months` ASC;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Years	Months	Number_of_Orders	MOVING_AVG
1	2016	9	4	4.0
2	2016	10	324	164.0
3	2016	12	1	109.67
4	2017	1	800	800.0
5	2017	2	1780	1290.0
6	2017	3	2682	1754.0
7	2017	4	2404	1916.5
8	2017	5	3700	2273.2
9	2017	6	3245	2435.17
10	2017	7	4026	2662.43

INSIGHT: Column `Number_of_Orders` shows the monthly orders which can be traced on a graph to show a clear uptrend

-Also, an ascending `Moving_Avg` shows a clear uptrend from September 2016, indicating a growing trend in the number of orders placed over the past years.
-However, an in-depth analysis, indicates a sharp decline between September 2018 and October 2018.

RECOMMENDATION: Based on the above insight, efforts can be made towards:
-Improving sales via incentivising sale executives.
-Business expansions to outlier cities
-Clearance sale campaigns during off seasons.
- Aggressive brand promotion

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

```
SELECT Q.*
FROM (SELECT T.*,MAX(`Number_of_Orders`) OVER(PARTITION BY `Years`) AS
`max_sales`
FROM(
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS`Years`,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS `Months`,
       COUNT(order_id) AS `Number_of_Orders`
FROM `target.orders`
GROUP BY `Years`,`Months`
ORDER BY `Years` ASC,`Months` ASC)AS `T`)AS `Q`
WHERE Q.`max_sales`= Q.`Number_of_Orders`
ORDER BY Q.`Years` ASC,Q.`Months` ASC;
```


Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Years	Months	Number_of_Orders	max_sales
1	2016	10	324	324
2	2017	11	7544	7544
3	2018	1	7269	7269

INSIGHT: In the above output Column “Years” displays years in an ascending order and the `Months’ Column displays month in which maximum no. of orders were placed against the respective years:

-Brazil's most festivals are celebrated during the spring and summer months of October to February.

-Rio Carnival, Oktoberfest, Cirio de Nazare, Natal Luz de Gramado and Bonfim Festival are some of the festivals that were observed in months in conformity with the peak orders placed in 2016, 2017 and 2018; thus, indicating seasonality.

-Festivities and carnival celebrations in Brazil are City centric and thus some cities may observe an influx of tourist. Consequently, observing a greater demand and customer traffic than other cities.

RECOMMENDATION: Company should temporarily shift focus on carnival cities during festival months by managing:
-Local inventories, so that timely replenishment from adjoining cities may handle a spike in demand in carnival cities during that time.
-Logistic Department, can temporarily transfer work force from adjoining states to fulfil logistic commitments.
-Local logistic partnerships can be done and alerted for the same as force multiplier.
-Festive celebratory offers commencing on carnivals dates and major Christian holydays.
-Launching of membership programs with salient features (free delivery/ guaranteed discounts to be availed later) during peak seasons can move sales during off seasons.

C. <u>During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)</u>

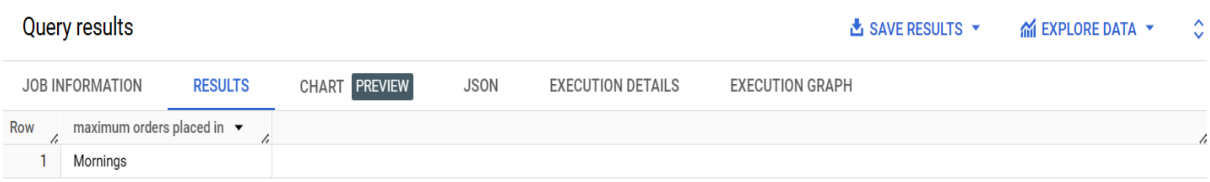
- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

/*Without Using Window Function*/

```
WITH A as
       (SELECT u.`Day_unit`,COUNT(u.`Day_unit`) AS `Count_units`
       FROM
       (SELECT t.*, IF(`Order_hrs` BETWEEN 0 and 6,"Dawn",
       IF(`Order_hrs` BETWEEN 7 and 12,"Mornings",
       IF(`Order_hrs` BETWEEN 13 and 18,"Afternoon",
       IF (Order_hrs BETWEEN 19 and 23,"Night","null")) )) AS `Day_unit`
       FROM
       (SELECT `order_id`,EXTRACT(HOUR FROM `order_purchase_timestamp`
       AT TIME ZONE'-03:00') AS `Order_hrs` /*CONVERTING UTC to BRT*/
       FROM `target.orders`
       ORDER BY order_id ASC) AS t) AS u
       GROUP BY u.Day_unit),

    B as
       (SELECT max(v.Count_units) AS `max_val`
       FROM
       (SELECT u.`Day_unit`,COUNT(u.`Day_unit`) AS `Count_units`
       FROM
       (SELECT t.*,
       IF(`Order_hrs` BETWEEN 0 and 6,"Dawn",
       IF(`Order_hrs` BETWEEN 7 and 12,"Mornings",
       IF(`Order_hrs` BETWEEN 13 and 18,"Afternoon",
       IF (Order_hrs BETWEEN 19 and 23,"Night","null")) )) AS `Day_unit`
       FROM
       (SELECT `order_id`,
             EXTRACT(HOUR FROM`order_purchase_timestamp`AT TIME ZONE'-
             03:00') AS `Order_hrs /*CONVERTING to BRT*/
       FROM `target.orders`
       ORDER BY order_id ASC) AS t) AS u
       GROUP BY u.`Day_unit`) AS v)


SELECT A.`Day_unit` AS `maximum orders placed in`
FROM A
INNER JOIN B ON A.`Count_units`= B.`max_val`;
```



INSIGHT: After converting the UTC to BRT it's been observed :
-Maximum order 're placed during the Morning hrs of 7-12 AM
-Followed by Afternoon hours as close second.

RECOMMENDATION: Based on the available information, it's recommended that:

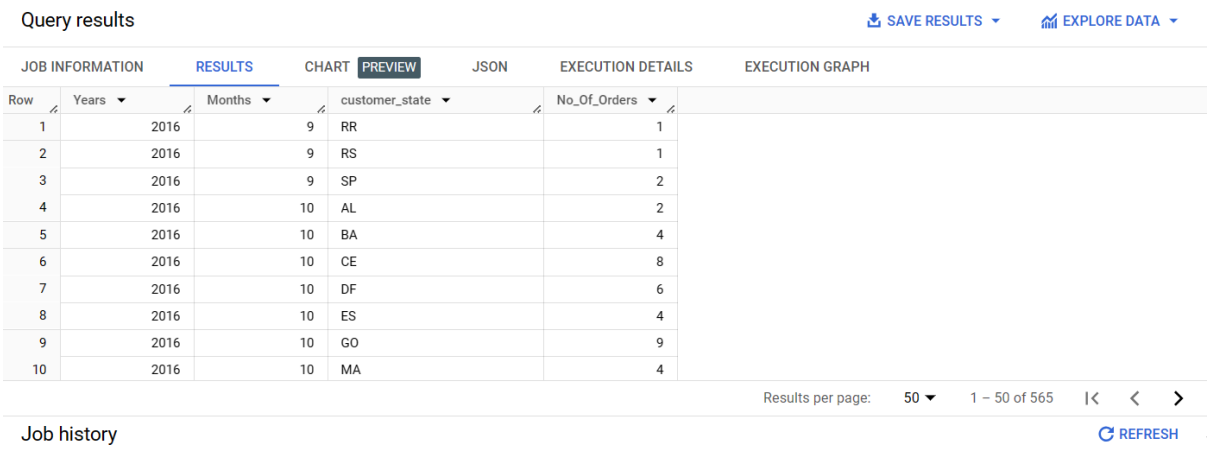
-Marketing department should book newspaper spaces and morning/afternoon slots in radio(office goers)and telemarketing (home makers).

- Push Notifications as Deal of the Day and Discount codes should be broadcasted during morning and afternoon hours.

III. Evolution of E-commerce orders in the Brazil region:

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

```
SELECT EXTRACT(YEAR FROM o.`order_purchase_timestamp`)`Years`,
       EXTRACT(MONTH FROM o.`order_purchase_timestamp`)`Months`,
       c.`customer_state`, COUNT(o.`order_id`) AS `No_Of_Orders`
FROM`target.customers`AS c
INNER JOIN `target.orders` AS o
ON c.`customer_id`= o.`customer_id`
GROUP BY `Years`,`Months`, c.`customer_state`
ORDER BY `Years` ASC,`Months` ASC, c.`customer_state` ASC;
```


Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Years	Months	customer_state	No_Of_Orders
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

Results per page: 50 1 – 50 of 565

Job history REFRESH

INSIGHT: Based on the figures we can analyse the performance of sales by comparing states on a month-on-month basis.

RECOMMENDATION: Based on the insight:

-A comprehensive study can be undertaken to formulate a performance index that can be instrumental in analysing State wise impact on orders during on and off-seasons.

- Further Data Mining will not only help us to resolve issues with states having poor order records but can also identify better performing states, thus replicating best practices.

- Consequently, customer survey can be conducted in states of particular concern for better customer experience.

- Furthermore, timely predictions based on such an index will add to robustness of the company's preparatory mechanisms.

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

```
SELECT customer_state,COUNT(DISTINCT customer_unique_id) AS
`Unique_Customers`
FROM [redacted].customers`
GROUP BY customer_state
ORDER BY customer_state ASC;
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	Unique_Customers
1	AC	77
2	AL	401
3	AM	143
4	AP	67
5	BA	3277
6	CE	1313
7	DF	2075
8	ES	1964
9	GO	1952
10	MA	726

Results per page: 50 1 – 27 of 27

INSIGHT: In the above output Column "customer_state" displays record of states in lexicographical order.

-E-commerce customers often make more than one accounts to avail single discount coupon multiple times.
-Unique customer ids help companies to identify their customers and to keep track of their transactions, interests, day to day interactions and inter customer relationships.

-A unique identifier can also help companies to comply with regulations, prevent fraud, and improve their data quality and accuracy.
-Unique customers ids not only remove duplicate data but also enables the companies to consult accurate information, it's an essential metric of any organisation's growth and performance.

RECOMMENDATION: Based on above insight:

- A transition from complex username and passwords to biometric login IDs (face and thumb impressions) can be recommended.
-Such a transition would not only be welcomed by customer at this day and age of flash deals and user-friendly interface but will also identify duplicate user accounts.

IV. Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.

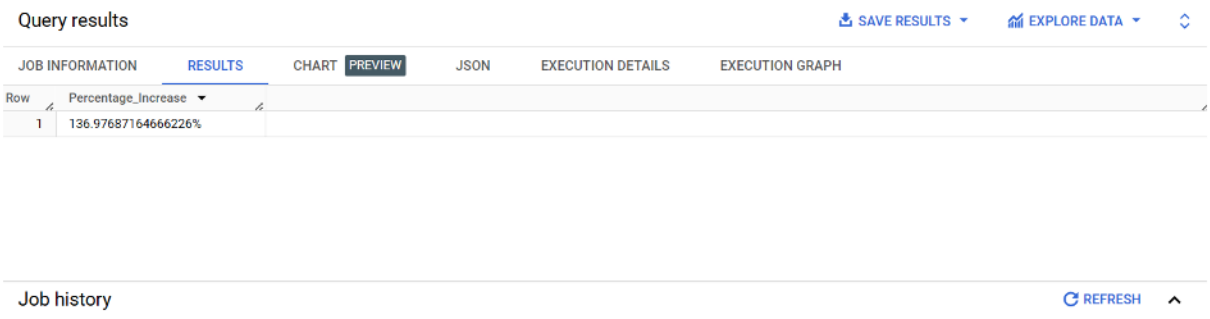
A. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
                /*Without Window Function*/
WITH A AS
      (SELECT IQ.`Year`,SUM(IQ.`Monthly_Cost`) AS `Jan-Aug_Cost`
      FROM(
      SELECT EXTRACT(YEAR FROM o.`order_purchase_timestamp`) as `Year`,
             EXTRACT(MONTH FROM o.`order_purchase_timestamp`) as `Month`,
             SUM(p.`payment_value`) as `Monthly_Cost`
      FROM [redacted].orders` AS o
      INNER JOIN `target.payments` AS p
      ON o.`order_id`=p.`order_id`
      GROUP BY `Year`,`Month`
      ORDER BY `Year` ASC,`Month` ASC) AS IQ
      WHERE IQ.`Year`BETWEEN 2017 and 2018 AND IQ.`Month` BETWEEN 1 AND 8
      GROUP BY IQ.`Year`),

      B AS
```

```
        (SELECT IQ.`Year`,SUM(IQ.`Monthly_Cost`) AS `Jan-Aug_Cost`
        FROM(
        SELECT EXTRACT(YEAR FROM o.`order_purchase_timestamp`) as `Year`,
               EXTRACT(MONTH FROM o.`order_purchase_timestamp`) as `Month`,
               SUM(p.`payment_value`) as `Monthly_Cost`
        FROM `[redacted].orders` AS o
        INNER JOIN `[redacted].payments` AS p
        ON o.`order_id`=p.`order_id`
        GROUP BY `Year`,`Month`
        ORDER BY `Year` ASC,`Month` ASC) AS IQ
        WHERE IQ.`Year`BETWEEN 2017 and 2018 AND IQ.`Month` BETWEEN 1 AND 8
        GROUP BY IQ.`Year`)

SELECT CONCAT(F.`Percentage_Increase`,"%") AS `Percentage_Increase`
FROM(
SELECT (T.`NEW`-T.`OLD`)*100 /T.`OLD` AS `Percentage_Increase`
FROM(
SELECT A.`Jan-Aug_Cost`as `NEW`,B.`Jan-Aug_Cost`AS `OLD`
FROM A
CROSS JOIN B
ORDER BY `OLD` ASC,`NEW` DESC
LIMIT 1) AS T) AS F;
```



INSIGHT: The percentage increase in the cost of orders from year 2017 to 2018 is 136.97687164666226% clearly indicated an upward trend thus emphasises a great demand of ecommerce among consumers.

RECOMMENDATIONS: The development in company policies which took place between 2017 and 2018 needs to be continued.

-The number of orders placed must be identified with the performance of the regional departments.

-Customers who cancelled their orders must be asked for their cause of disapproval to improve sales and quality of service.

-Feasibility Studies can be conducted consequently deepening the reach of the company in states and respective cities.

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

```
SELECT c.customer_state,sum(i.`price`)AS `Total_Value`,
       avg(i.`price`) AS `Avg_Value`

FROM `[redacted]c.customers` as c
LEFT JOIN `target.orders` as o ON c.`customer_id`=o.`customer_id`
LEFT JOIN `target.order_items` as i ON o.`order_id`= i.`order_id`
GROUP BY c.customer_state
ORDER BY c.customer_state ASC;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	Total_Value	Avg_Value
1	AC	15982.94999999...	173.7277173913...
2	AL	80314.81000000...	180.8892117117...
3	AM	22356.84000000...	135.4959999999...
4	AP	13474.29999999...	164.3207317073...
5	BA	511349.9900000...	134.6012082126...
6	CE	227254.7099999...	153.7582611637...
7	DF	302603.9399999...	125.7705486284...
8	ES	275037.3099999...	121.9137012411...
9	GO	294591.9499999...	126.2717316759...
10	MA	119648.2199999...	145.2041504854...

Results per page: 50 1 – 27 of 27

Job history REFRESH

INSIGHT: In the above output Column “customer_state” displays record of states in lexicographical order.

-State `SP’ has the highest value of order price but the lowest average value of order price.

-Thus, it’s observed that states with greater average value of order price perhaps may have consumers with greater buying power but such states not necessarily account for greater revenues for the company.
-States with higher value of order price account for greater no. of orders for the company, consequently greater revenues.

RECOMMENDATION: Based on above mentioned points:

- Special resource allocation should be done for states with higher value of order price.

-A socio-economic and demographical study should be conducted to identify the reasons behind the inadequacy of states with lower value of order price for better insight and growth potential.

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

```
SELECT c.customer_state, SUM(i.`freight_value`)AS `Total_Freight_Value`,
AVG(i.`freight_value`) AS `Avg_Freight_Value`
FROM `[redacted].customers` as c
LEFT JOIN `[redacted].orders` as o
ON c.`customer_id`=o.`customer_id`
LEFT JOIN `[redacted].order_items` as i
ON o.`order_id`= i.`order_id`
GROUP BY c.customer_state
ORDER BY c.customer_state ASC;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	Total_Freight_Value	Avg_Freight_Value
1	AC	3686.749999999...	40.07336956521...
2	AL	15914.58999999...	35.84367117117...
3	AM	5478.889999999...	33.20539393939...
4	AP	2788.500000000...	34.00609756097...
5	BA	100156.6799999...	26.36395893656...
6	CE	48351.58999999...	32.71420162381...
7	DF	50625.49999999...	21.04135494596...
8	ES	49764.59999999...	22.05877659574...
9	GO	53114.97999999...	22.76681525932...
10	MA	31523.77000000...	38.25700242718...

Results per page: 50 | 1 – 27 of 27

INSIGHT: In the above output Column “customer_state” displays record of states in lexicographical order.

- State `SP’ has the highest total freight value but the lowest average freight value.

-State `RR' has the highest average freight value but the lowest total freight value.

-Higher average freight value is a good trade off if it comes at the cost of lower total freight cost.

- Freight charges are influenced by various factors such as weight, dimensions, distance, mode of transport and any extra services like tracking or insurance (optional).

RECOMMENDATIONS: Based on the above insight:

- Customers should be encouraged to buy in bulk by launching offers such as `Buy One Get 50% Off` and `Buy Two Get Three`.

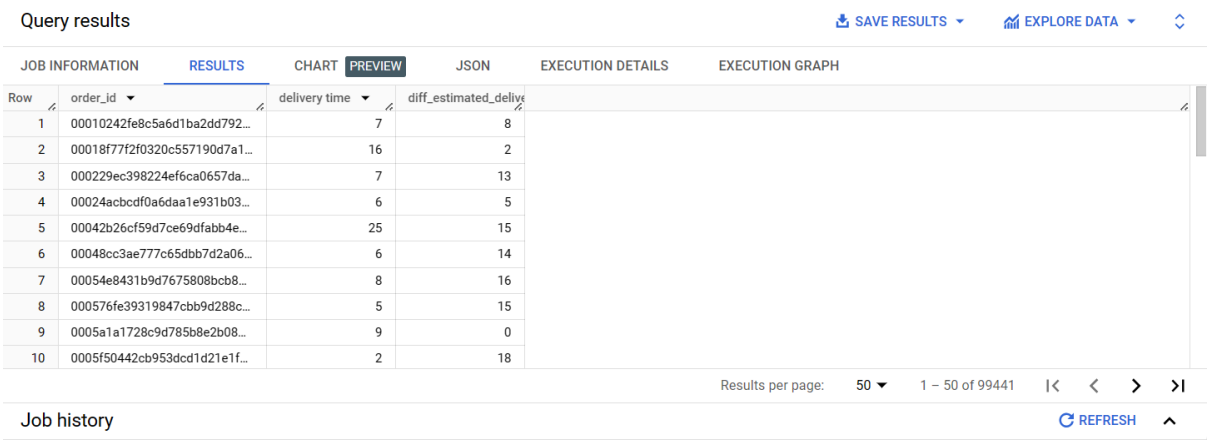
- Negotiating with logistic partners for shipping on off peak days of the week.

- Identifying smaller packing techniques that may save dimensions and consolidate greater room for more shipments.

V. Analysis based on sales, freight and delivery time.

A. 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.

```
SELECT `order_id`,
TIMESTAMP_DIFF(`order_delivered_customer_date`,`order_purchase_timestamp`,DA
Y) AS `delivery time`,
TIMESTAMP_DIFF(`order_estimated_delivery_date`,`order_delivered_customer_dat
e`,DAY) AS `diff_estimated_delivery `
FROM`[redacted].orders`
ORDER BY `order_id` ASC;
```



Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	order_id	delivery time	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

Results per page: 50 | 1 – 50 of 99441

Job history | REFRESH

INSIGHT: In the above output Column `order_id' displays ids. Column `delivery time' displays no. of days taken to deliver each order from the order's purchase date and Column `diff_estimated_delivery' displays difference (in days) between the estimated & actual delivery date of an order:

- Estimated delivery dates are calculated by adding the maximum handling and in-transit days.

- Delivery time is inversely proportional to difference (in days) between the estimated & actual delivery date of an order.

RECOMMENDATIONS: Based on the above insight:

- Estimated delivery dates should be set after considering the historical accounts of customer's geolocations and the response time of the logistical partner catering to them.

- It's recommended that dispatching of orders to the logistic partners must be done at the earliest prioritising orders with shorter estimated delivery dates.

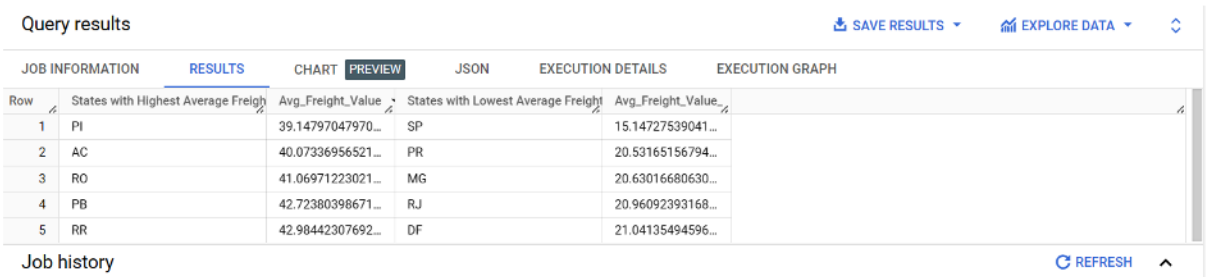
- Negotiating with logistic partners for shipping on off peak days of the week to avoid any accumulation of pending orders ready to be shipped.

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

```
WITH   A AS(
       SELECT a.*,DENSE_RANK() OVER(ORDER BY a.`Avg_Freight_Value`ASC)
AS`TOP_5A`
       FROM(
       SELECT T.*, DENSE_RANK() OVER(ORDER BY T.`Avg_Freight_Value`DESC)
AS`TOP_5F`
       FROM (
       SELECT c.customer_state as `States with Highest Average Freight
Value`,avg(i.`freight_value`) AS `Avg_Freight_Value`
       FROM `[redacted].customers` as c
       LEFT JOIN `[redacted].orders` as o ON c.`customer_id`=o.`customer_id`
       LEFT JOIN `[redacted].order_items` as i ON o.`order_id`= i.`order_id`
       GROUP BY c.`customer_state`)AS T) AS a
       WHERE a.`TOP_5F`<=5
       ORDER BY a.`TOP_5F` ASC),

 B AS  (SELECT a.*
       FROM(
       SELECT T.*, DENSE_RANK() OVER(ORDER BY T.`Avg_Freight_Value`ASC)
AS`BOTTOM_5F`
       FROM (
       SELECT c.customer_state as `States with Lowest Average Freight
Value`,avg(i.`freight_value`) AS `Avg_Freight_Value`
       FROM `[redacted].customers` as c
       LEFT JOIN `[redacted].orders` as o ON c.`customer_id`=o.`customer_id`
       LEFT JOIN `[redacted].order_items` as i ON o.`order_id`= i.`order_id`
       GROUP BY c.`customer_state`)AS T) AS a
       WHERE a.`BOTTOM_5F`<=5
       ORDER BY a.`BOTTOM_5F` ASC)

SELECT A.`States with Highest Average Freight Value`,A.`Avg_Freight_Value`,
       B.`States with Lowest Average Freight Value`,B.`Avg_Freight_Value`as
`Avg_Freight_Value_`
FROM A
INNER JOIN B
ON A.`TOP_5A`=B.`BOTTOM_5F`
ORDER BY A.`TOP_5A`;
```



Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	States with Highest Average Freigh	Avg_Freight_Value	States with Lowest Average Freight	Avg_Freight_Value_
1	PI	39.14797047970...	SP	15.14727539041...
2	AC	40.07336956521...	PR	20.53165156794...
3	RO	41.06971223021...	MG	20.63016680630...
4	PB	42.72380398671...	RJ	20.96092393168...
5	RR	42.98442307692...	DF	21.04135494596...

Job history

REFRESH

INSIGHT: States with Highest Freight Value and States with Lowest Freight Value are arranged in their respective columns with increasing order of their average delivery time:

- State `SP' has the lowest average freight value but has the highest total freight.

-State `RR' has the highest average freight value but the lowest total freight value.

-Higher average freight value is a good trade at the cost of lower total freight cost.

- Freight charges are influenced by various factors such as weight, dimensions, distance, mode of transport and any extra services like tracking or insurance (optional).

RECOMMENDATION: Efforts must be made to choose tonnage over dunnage:

- Encouraging customers to buy in bulk.

- Identifying smaller packing techniques that may save dimensions and consolidate greater room for more shipments.

C. <u>Find out the top 5 states with the highest & lowest average delivery time</u>

```
WITH A AS(
       SELECT u.*,DENSE_RANK() OVER(ORDER BY u.`Avg_Delivery_Time` ASC)
AS`RAN_REV`
       FROM(
       SELECT t.*, DENSE_RANK() OVER(ORDER BY t.`Avg_Delivery_Time` DESC)
AS`RAN_A`
       FROM(
       SELECT c.`customer_state` AS `States with longest average delivery
time`,
       AVG(TIMESTAMP_DIFF(o.`order_delivered_customer_date`,o.`order_purchas
e_timestamp`, DAY))AS `Avg_Delivery_Time`
       FROM `[redacted].customers`AS c
       LEFT JOIN `[redacted].orders`AS o
       ON c.`customer_id`=o.`customer_id`
       GROUP BY `States with longest average delivery time`)AS t)AS u
       WHERE u.`RAN_A`<=5
```

```
       ORDER BY `RAN_REV`ASC),

 B AS(  SELECT u.*
       FROM(
       SELECT t.*, DENSE_RANK() OVER(ORDER BY t.`Avg_Delivery_Time` ASC)
AS`RAN_B`
       FROM(
       SELECT c.`customer_state` AS `States with shortest average delivery
time`,
       AVG(TIMESTAMP_DIFF(o.`order_delivered_customer_date`,o.`order_purchas
e_timestamp`, DAY))AS `Avg_Delivery_Time`
       FROM [redacted]t.customers`AS c
       LEFT JOIN [redacted].orders`AS o
       ON c.`customer_id`=o.`customer_id`
       GROUP BY `States with shortest average delivery time`)AS t)AS u
       WHERE u.`RAN_B`<=5)

SELECT A.`States with longest average delivery time`,A.`Avg_Delivery_Time`,
       B.`States with shortest average delivery time`,B.`Avg_Delivery_Time`
AS `Avg_Delivery_Time_`
FROM A
INNER JOIN B
ON A.`RAN_REV`=B.`RAN_B`
ORDER BY A.`RAN_REV` ASC;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	States with longest average deliver	Avg_Delivery_Time	States with shortest average delive	Avg_Delivery_Time_
1	PA	23.31606765327...	SP	8.298061489072...
2	AL	24.04030226700...	PR	11.52671135486...
3	AM	25.98620689655...	MG	11.54381329810...
4	AP	26.73134328358...	DF	12.50913461538...
5	RR	28.97560975609...	SC	14.47956019171...

Job history

REFRESH

INSIGHT: States with longest average delivery time and States with shortest average delivery time are arranged in their respective columns in order of increasing average delivery time:

- State `SP' has the shortest average delivery time and also from the previously mentioned insights it was observed that State `SP' has also the lowest average freight value but has the highest total freight value.

-State `RR' has the longest average delivery time and also from the previously mentioned insights it was observed that

State `RR' has the highest average freight value but the lowest total freight value.

-It's been observed that shorter average delivery time is a trade off at the cost of higher total freight value.

RECOMMENDATIONS: Based on above Insights its recommended:

-For customer satisfaction, it's imperative to strike an even balance between lower cost of freight value and higher cost of average freight value to get an optimal delivery time.

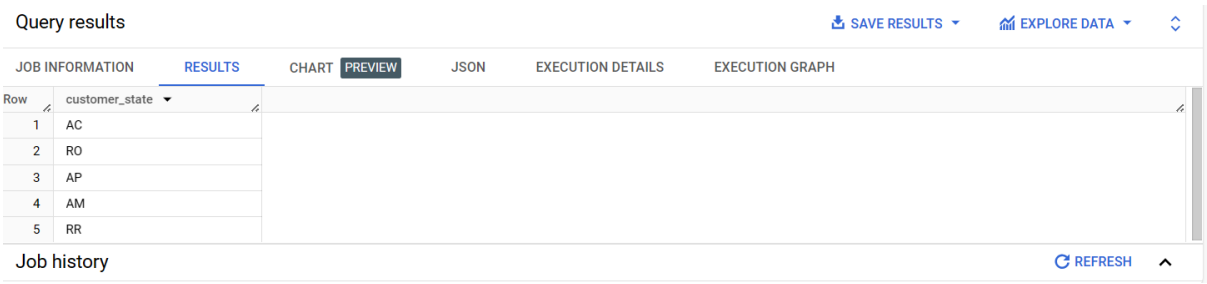
- Sorting and separating different types or categories of orders can eliminating unnecessary processes and may quicken the delivery process at Sellers location.

-Further, automating the routine processes like sorting and labelling can also save a lot of time and fasten the process.

D. <u>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.</u>

```
SELECT r.customer_state
FROM(
SELECT q.customer_state,q.`estimated_delivery_time`-q.`delivery_time` AS
`Diff`
FROM (
SELECT t.customer_state,
 AVG(TIMESTAMP_DIFF(t.`order_delivered_customer_date`,t.`order_purchase_t
imestamp`,DAY)) AS`delivery_time`,
 AVG(TIMESTAMP_DIFF(t.`order_estimated_delivery_date`,t.`order_purchase_t
imestamp`,DAY)) AS`estimated_delivery_time`
FROM(
SELECT
c.customer_state,o.`order_purchase_timestamp`,o.`order_delivered_customer
_date`,o.`order_estimated_delivery_date`
FROM [redacted].customers` AS c
```

```
INNER JOIN  `target.orders` AS o
ON c.`customer_id`=o.`customer_id` and o.order_status="delivered") AS t
GROUP BY customer_state) AS q
ORDER BY `Diff` DESC
LIMIT 5)AS r
ORDER BY r. `Diff` DESC ;
```



INSIGHT: Top 5 States where the where the order delivery is really fast as compared to the estimated date of delivery is displayed. Where State `AC' has fastest order delivery and state `RR' have slowest in that order compared to the estimated delivery date.

RECOMMENDATION: Based on the above insight:

- Estimated delivery dates should be set after considering the historical accounts of customer's geolocations and the response time of the logistical partner catering to them.

-Customers should be notified in real time about the location of their orders in transit.

VI. Analysis based on the payments:

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

```
SELECT EXTRACT(YEAR FROM o.`order_purchase_timestamp`) AS `YEARS`,
       EXTRACT(MONTH FROM o.`order_purchase_timestamp`) AS `MONTHS`,
       p.`payment_type`, count(p.`payment_type`) AS `Count_type`
FROM [redacted]t.orders` AS o
INNER JOIN `[redacted].payments`AS p
ON o.`order_id`=p.`order_id`
GROUP BY `YEARS`,`MONTHS`,p.`payment_type`
ORDER BY `YEARS`,`MONTHS`,p.`payment_type`;
```

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	YEARS	MONTHS	payment_type	Count_type
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	583
9	2017	1	debit_card	9
10	2017	1	voucher	61

Results per page: 50 | 1 – 50 of 90

Job history | REFRESH

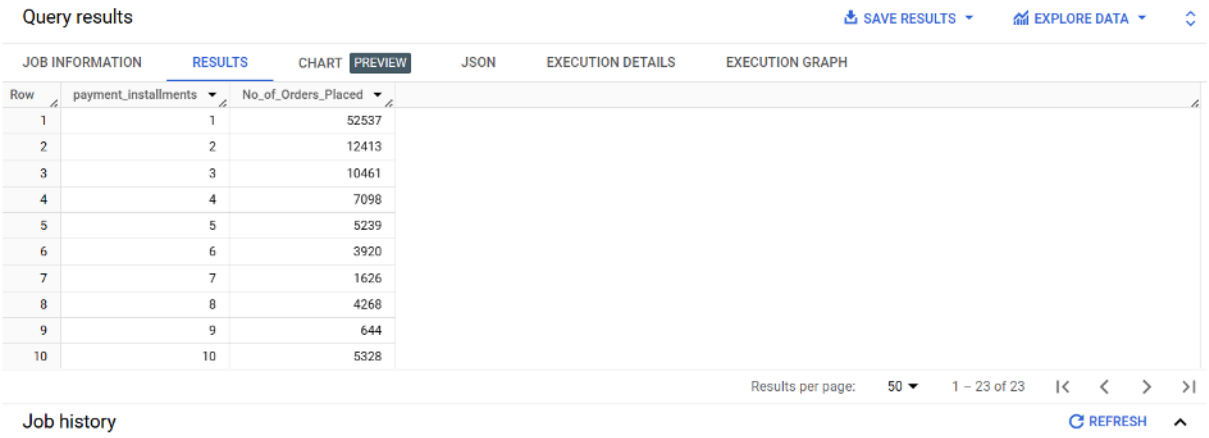
INSIGHT: The month-on-month no. of orders placed using different payment types is arranged along with their respective payment type from September 2016 to October 2018.

RECOMMENDATION: Customers should be encouraged to spend more:
- Partnership deals with leading Banks and Credit Card Companies to launch frequent offers and attractive deals.
-Development of secure payment gateways, safeguarding. customer payment details to promote safe and secure online modes of transaction making it seamless operation.
-Swift reimbursements and resolution of complaints.
-Zero EMI offers by partner banks.
-Buy now pay later.

B. Find the no. of orders placed on the basis of the payment instalments that have been paid.

```
SELECT `payment_installments`,
       COUNT(`order_id`) AS `No_of_Orders_Placed`

FROM `[redacted].payments`
WHERE `payment_value`>0 AND `payment_installments`>0
GROUP BY `payment_installments`
ORDER BY  `payment_installments`ASC;
```



Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	payment_installments	No_of_Orders_Placed
1	1	52537
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

Results per page: 50 | 1 – 23 of 23

Job history | REFRESH

INSIGHT: Number of orders placed and that have been paid are arranged in the ascending order of the payment instalments are displayed; whereas, no. of orders placed with 0 instalments are 2.

SUGGESTION: Customers should be encouraged to spend more :

- Partnership deals with leading Banks and Credit Card. Companies to launch frequent offers and attractive deals.
-Zero EMI offers by partner banks.
-Buy now pay later options can be included.