HANUMANTH RAM T **BUSINESS_CASE**

Problem Statement:

Assuming you are a data analyst/ scientist at an Anonymous Mass-market retail Company, you have been assigned thetask of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

The data was available in 8 csv files:

1. **customers.csv**
2. **sellers.csv**
3. **order_items.csv**
4. **geolocation.csv**
5. **payments.csv**
6. **reviews.csv**
7. **orders.csv**
8. **products.csv**

The **customers.csv** 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 consumer's location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The **sellers.csv** contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller's location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The **order_items.csv** 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 the Company
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The **geolocations.csv** contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code
geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The **payments.csv** 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 (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

The **orders.csv** 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

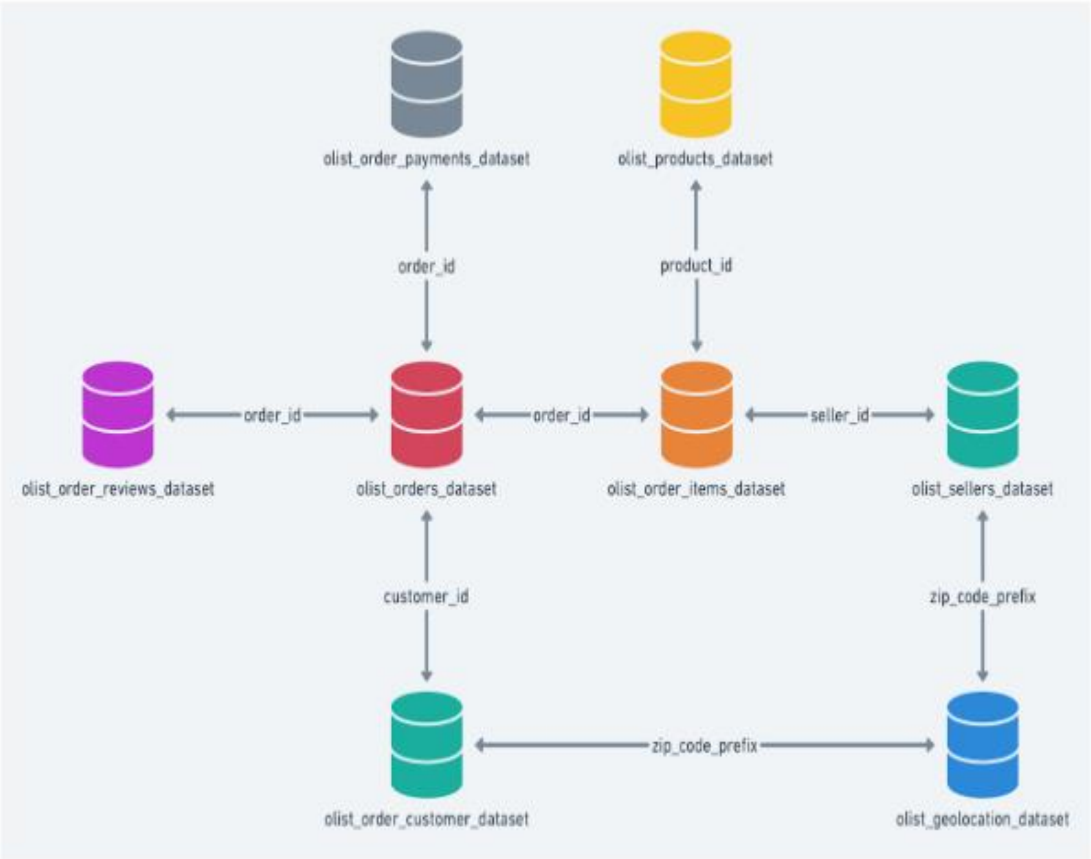
The **reviews.csv** contain following features:

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 a 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.csv** 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

Dataset schema:

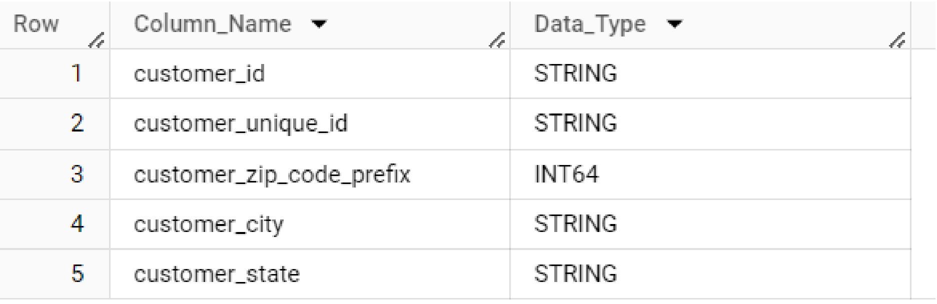


1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
    1. Data type of all columns in the "customers" table.

Query:

```
SELECT `Column_Name`, `Data_Type`
FROM ` INFORMATION_SCHEMA.COLUMNS` where table_name='customers'
```

Output:



Row	Column_Name	Data_Type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Inference:

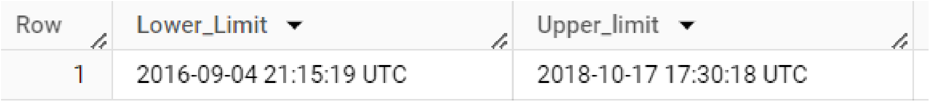
There are 5 columns in the given customers table namely customer_id with data type string, customer_unique_id with data type string, customer_zip_code_prefix with data type integer, customer_city with data type string, customer_state with data type string.

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

Query:

```
SELECT min(`order_purchase_timestamp`) as `Lower_Limit`,
max(`order_purchase_timestamp`) as `Upper_limit`
FROM  `orders`
```

Output:



Row	Lower_Limit	Upper_limit
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Inference:

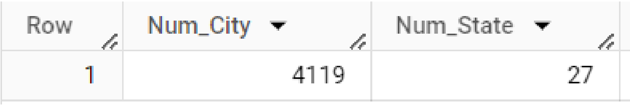
The orders were placed between 21:15 UTC in 4th of September 2016 and 17:30 UTC in 17th October 2018.

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

Query:

```
SELECT COUNT(DISTINCT C.CUSTOMER_CITY) as Num_City,
COUNT(DISTINCT C.CUSTOMER_STATE) as Num_State
FROM `orders` as O
LEFT JOIN `customers` as C
ON O.CUSTOMER_ID=C.CUSTOMER_ID
```

Output:



Row	Num_City	Num_State
1	4119	27

Inference:

There were customers over 4119 Cities and 27 States who had placed their Orders in the given Period.

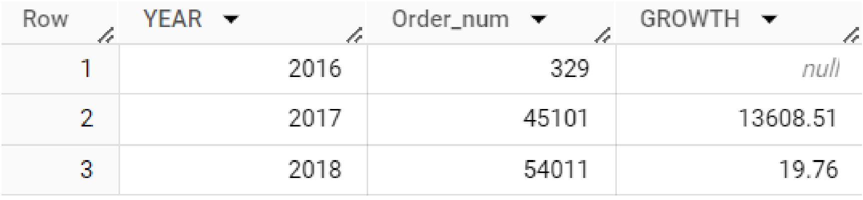
2. In-depth Exploration:

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

Query:

```
WITH A as
(SELECT *, EXTRACT(YEAR FROM ORDER_PURCHASE_TIMESTAMP) as
`Year`
FROM `orders`),
B as
(SELECT A.YEAR, COUNT(A.ORDER_ID) as Order_num
FROM A
GROUP BY A.YEAR),
C as
(SELECT *, LAG(B.Order_num) OVER(order by Year asc) as `LAG`
FROM B
ORDER BY YEAR)
SELECT C.YEAR,C.Order_num, Round((C.Order_num-`LAG`)*100/`LAG`,2) as
`GROWTH`
FROM C
```

Output:



Row	YEAR	Order_num	GROWTH
1	2016	329	*null*
2	2017	45101	13608.51
3	2018	54011	19.76

Inference:

The number of Orders grew Enormously from the Year 2016 to the Year 2017 with 13608.51% increase in Growth Rate. But since the Data is from September of 2016, and only 4 months' orders were recorded for 2016, hence we are noticing huge growth rate value. The number of Orders grew Significantly from the Year 2017 to Year 2018 with 19.76% increase in Growth Rate. Considering the Data were till October 2018, there could have been more orders placed in the Year 2018, could have led to more growth rate value. Yes, there is a growing trend of number of orders placed in the given period.

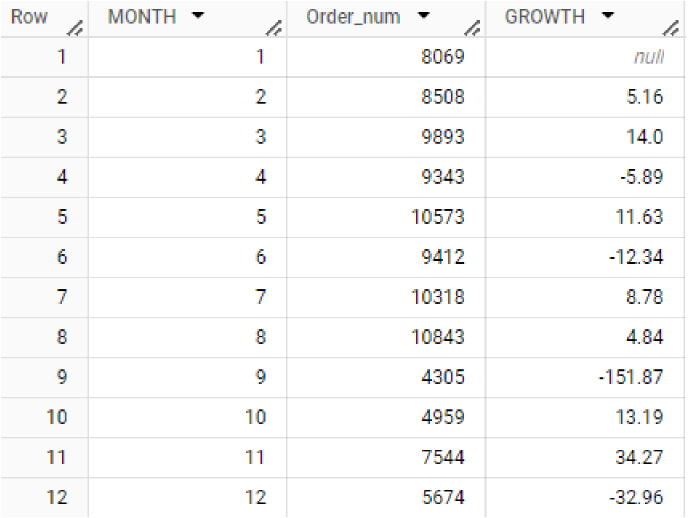
2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Query:

```
WITH A as
(SELECT *, EXTRACT(MONTH FROM ORDER_PURCHASE_TIMESTAMP) as
`Month`
FROM `orders`),
B as
(SELECT A.MONTH, COUNT(A.ORDER_ID) as Order_num
FROM A
GROUP BY A.MONTH),
C as
```

```
(SELECT *, LAG(B.Order_num) OVER(order by MONTH asc) as `LAG`
FROM B
ORDER BY MONTH)
SELECT C.MONTH,C.Order_num, Round((C.Order_num-
`LAG`)*100/C.Order_num,2) as `GROWTH`
FROM C
```

Output:



Row	MONTH	Order_num	GROWTH
1	1	8069	null
2	2	8508	5.16
3	3	9893	14.0
4	4	9343	-5.89
5	5	10573	11.63
6	6	9412	-12.34
7	7	10318	8.78
8	8	10843	4.84
9	9	4305	-151.87
10	10	4959	13.19
11	11	7544	34.27
12	12	5674	-32.96

Inference:

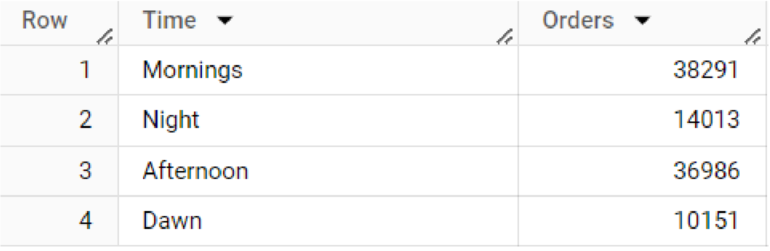
August had the most number of orders and the number of orders drops significantly in September, 151.87% decrease compared to August. The months of September, October, November, December have relatively less orders compared to the rest of the months. Yes, there is a monthly seasonality persist in the number of orders placed.

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

Query:

```
With O as
(SELECT extract(Hour FROM DATETIME(order_purchase_timestamp,"-3:00")) as
Order_hr
FROM `orders`),
T as
(SELECT
Order_hr,IF(Order_hr<7,"Dawn",IF(Order_hr<13,"Mornings",IF(Order_hr<19,"After
noon","Night"))) as `Time`
FROM O)
SELECT T.`Time`,COUNT(T.Time) as `Orders`
FROM T
GROUP BY T.`Time`
```

Output:



Row	Time	Orders
1	Mornings	38291
2	Night	14013
3	Afternoon	36986
4	Dawn	10151

Inference:

The Brazil Customers mostly placed their orders in the Mornings and then followed by Afternoon. Least orders were placed in Dawn. Notifications can be given to the customers in the Mornings and Afternoons to promote the customers to order in the company.

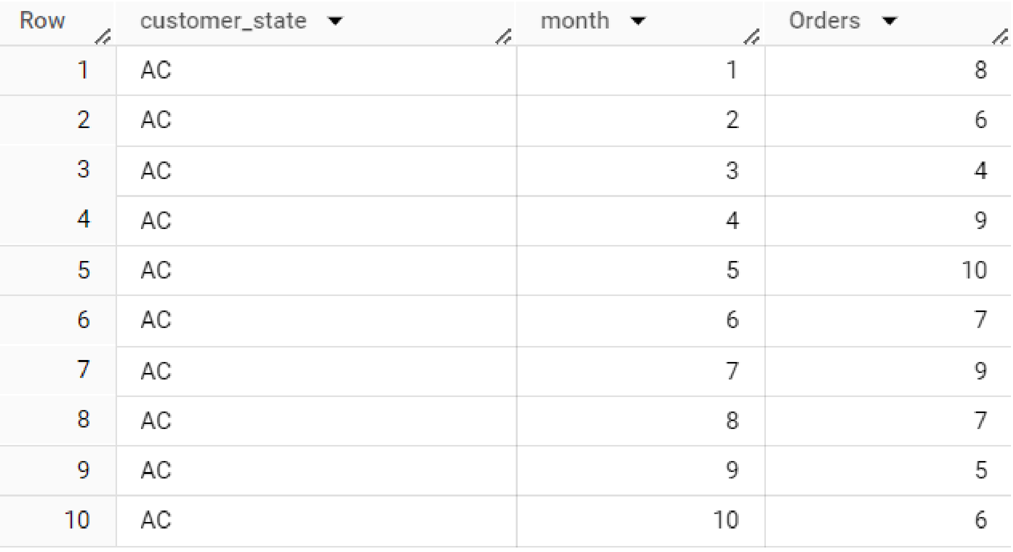
3. Evolution of E-commerce orders in the Brazil region:

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

Query:

```
SELECT C.customer_state,extract(Month FROM
DATETIME(O.order_purchase_timestamp,"-3:00")) as `month`,
COUNT(O.order_id) as Orders
FROM `orders` as O
LEFT JOIN `customers` as C
ON O.customer_id=C.customer_id
GROUP BY C.customer_state, month
Order by C.customer_state, month asc;
```

Output:



Row	customer_state	month	Orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Inference:

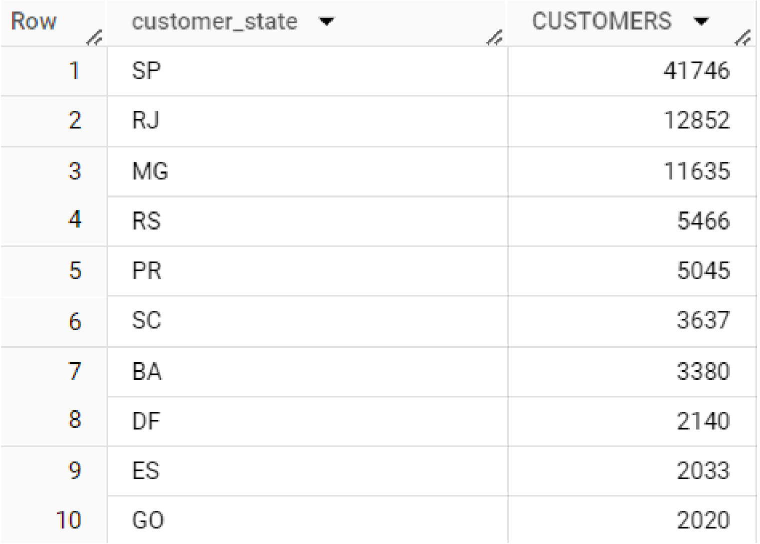
The month on month no. of orders placed in each state is displayed. These data can also be analysed to understand the State specific seasonality of purchases in Brazil.

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

Query:

```
SELECT customer_state, Count(DISTINCT customer_id) as `CUSTOMERS`
FROM `customers`
GROUP BY customer_state
ORDER BY `CUSTOMERS` desc;
```

Output:



Row	customer_state	CUSTOMERS
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Inference:

There are most number of Customers from the SP State of Brazil with 41746 customers and Least number of Customers in RR State in Brazil with 46 customers. Company can prioritize in taking steps to improve the number of customers from States which accounted with less customers to increase the Profit margin.

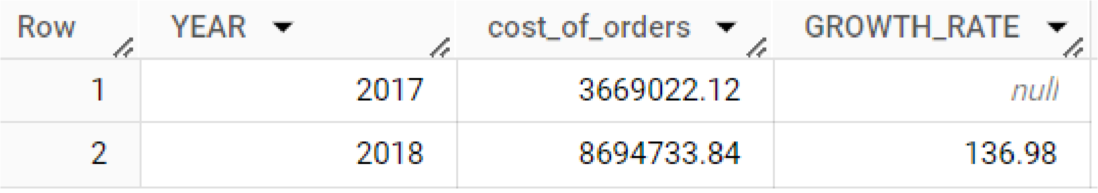
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

Query:

```
WITH A as (
SELECT EXTRACT(YEAR from O.order_purchase_timestamp) as `Year`,
ROUND(SUM(payment_value),2) as cost_of_orders
FROM `orders` as O
INNER JOIN `payments` as P
ON O.order_id=P.order_id
WHERE EXTRACT(MONTH from O.order_purchase_timestamp) not in
(9,10,11,12)
GROUP BY `Year`),
B as (
SELECT *,LAG(cost_of_orders) OVER(order by Year ASC) as `LAG` from A)
SELECT  YEAR,cost_of_orders,ROUND((cost_of_orders-`LAG`)*100/`LAG`,2)  as
`GROWTH_RATE` from B
ORDER BY YEAR ASC
```

Output:



Row	YEAR	cost_of_orders	GROWTH_RATE
1	2017	3669022.12	*null*
2	2018	8694733.84	136.98

Inference:

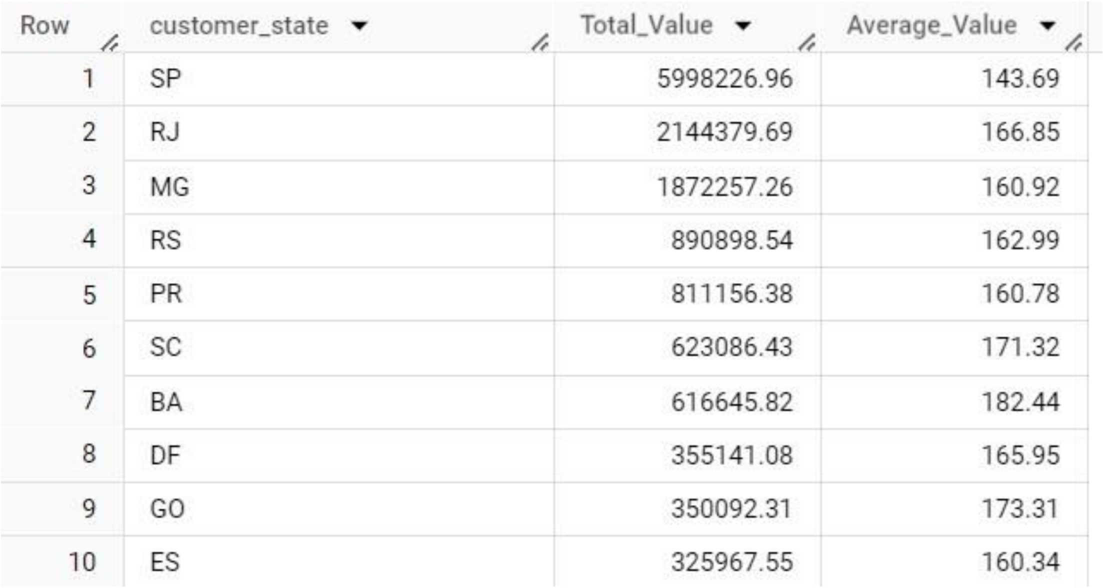
There was an Increase in Cost of Orders from the Year 2017 to the Year 2018, nearly 136.98 %, by only considering the months from January to August.

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

Query:

```
WITH A as (
SELECT O.customer_id, O.order_id, SUM(P.payment_value) as `order_price`
FROM `orders` as O
LEFT JOIN `payments` as P
ON O.order_id=P.order_id
GROUP BY O.customer_id, O.order_id)
SELECT C.customer_state, ROUND(SUM(order_price),2) as Total_Value,
ROUND(AVG(order_price),2) as Average_Value
from A
INNER JOIN `customers` as C
ON A.customer_id=C.customer_id
GROUP BY C.customer_state
ORDER BY Total_Value desc;
```

Output:



Row	customer_state	Total_Value	Average_Value
1	SP	5998226.96	143.69
2	RJ	2144379.69	166.85
3	MG	1872257.26	160.92
4	RS	890898.54	162.99
5	PR	811156.38	160.78
6	SC	623086.43	171.32
7	BA	616645.82	182.44
8	DF	355141.08	165.95
9	GO	350092.31	173.31
10	ES	325967.55	160.34

Inference:

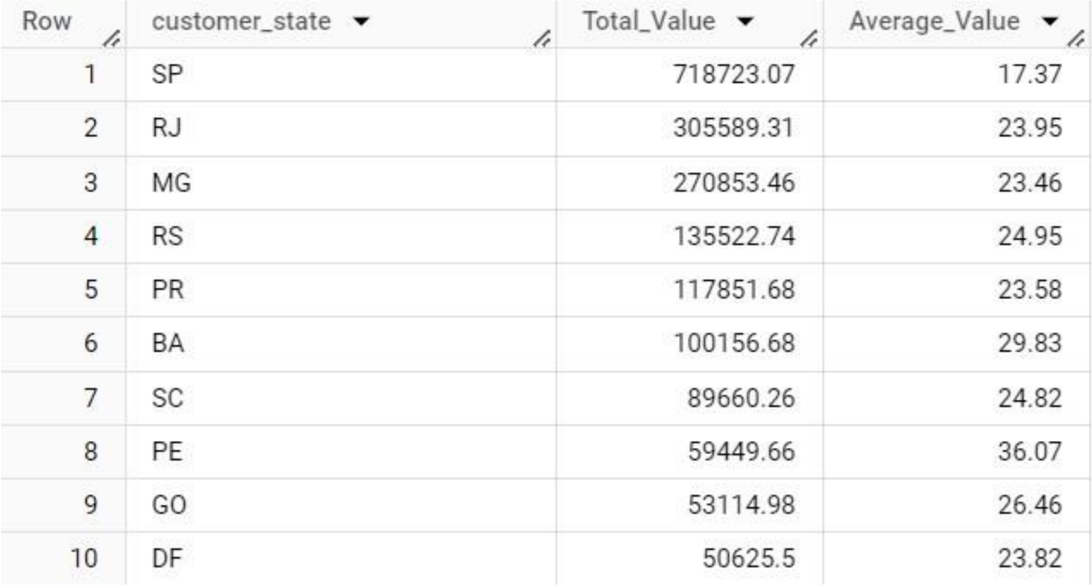
The company acquired most revenue from the SP State, nearly 6 Million. Followed by RJ State with 2.1 Million. PB State has the most Average value of order price. Company can prioritize in taking steps to improve the number of customers from States which accounted for less value.

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

Query:

```
WITH A as (
SELECT O.customer_id, O.order_id, SUM(OI.freight_value) as `freight_value`
FROM `orders` as O
LEFT JOIN `order_items` as OI
ON O.order_id=OI.order_id
GROUP BY O.customer_id, O.order_id)
SELECT C.customer_state, ROUND(SUM(freight_value),2) as Total_Value,
ROUND(AVG(freight_value),2) as Average_Value
from A
INNER JOIN `customers` as C
ON A.customer_id=C.customer_id
GROUP BY C.customer_state
ORDER BY Total_Value desc;
```

Output:



Row	customer_state	Total_Value	Average_Value
1	SP	718723.07	17.37
2	RJ	305589.31	23.95
3	MG	270853.46	23.46
4	RS	135522.74	24.95
5	PR	117851.68	23.58
6	BA	100156.68	29.83
7	SC	89660.26	24.82
8	PE	59449.66	36.07
9	GO	53114.98	26.46
10	DF	50625.5	23.82

Inference:

SP state inquired the most freight charges, logistics optimization could help in reducing freight costs and improve the turnover of the company.

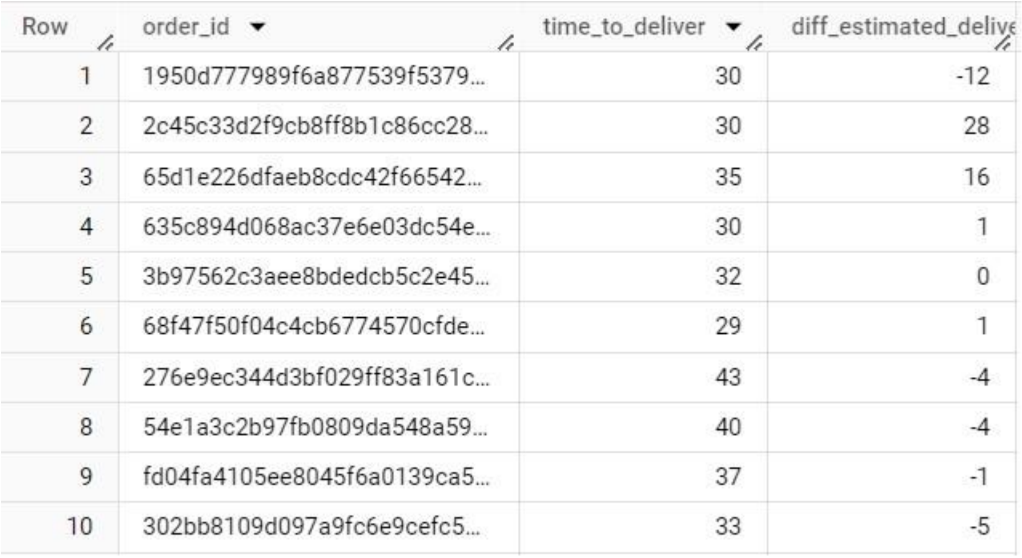
5. Analysis based on sales, freight and delivery time.

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.

Query:

```
SELECT order_id,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day)
as diff_estimated_delivery
FROM `orders`
```

Output:



Row	order_id	time_to_deliver	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

Inference:

The time taken to deliver the purchased product and the difference between estimated date of delivery and the date of delivery was computed as time_to_deliver and diff_estimated_delivery. The positive value in diff_estimated_delivery indicates products delivered before the estimated time, 0 indicates products delivered at the time of estimated time and negative values indicates products delivered later than the estimated time. The company can work on logistics and warehousing in delivering the product much before the estimated time to increase their reputation between customers.

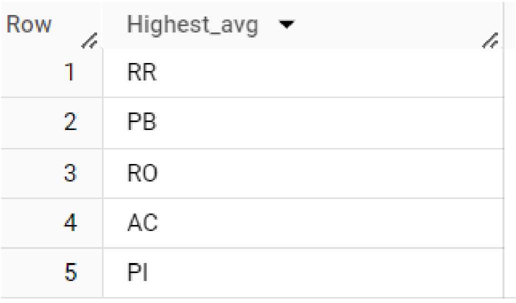
2. Find out the top 5 states with the highest & lowest average freight value.

Query:

```
WITH A as (
SELECT O.customer_id, O.order_id, SUM(OI.freight_value) as `freight_value`
FROM `orders` as O
LEFT JOIN `order_items` as OI
ON O.order_id=OI.order_id
GROUP BY O.customer_id, O.order_id),
B as (
SELECT C.customer_state, ROUND(SUM(freight_value),2) as Total_Value,
ROUND(AVG(freight_value),2) as Average_Value
from A
INNER JOIN `customers` as C
ON A.customer_id=C.customer_id
GROUP BY C.customer_state),
C as (
SELECT customer_state as Highest_avg
FROM B
ORDER BY Average_Value desc
LIMIT 5),
D as (
SELECT customer_state as Lowest_avg
FROM B
ORDER BY Average_Value asc
LIMIT 5)
```

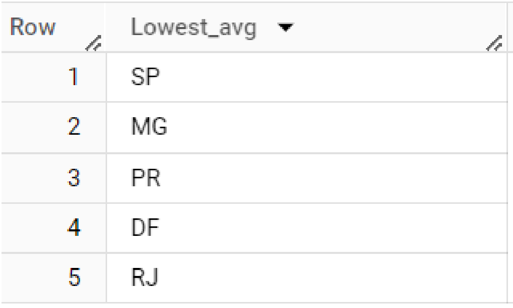
Output:

```
SELECT * from C
```



Row	Highest_avg
1	RR
2	PB
3	RO
4	AC
5	PI

```
SELECT * from D
```



Row	Lowest_avg
1	SP
2	MG
3	PR
4	DF
5	RJ

Inference:

The States of Top 5 Highest and Lowest Average Freight Charges are computed. The States with most value of freight charges have the lowest average frieght charge per Orders. States with High average Freight value can have their Freight value be reduced with interventions such as logistic vehicle routing and proper planning. Understand sitable Locations for warehousing can also help in reducing freight charges.

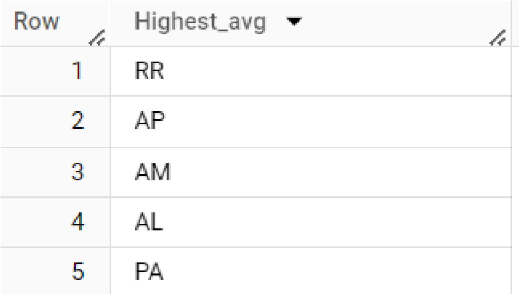
3. Find out the top 5 states with the highest & lowest average delivery time.

Query:

```
WITH A as (
SELECT C.customer_state,
ROUND(AVG(DATE_DIFF(O.order_delivered_customer_date,O.order_purchase_timestamp,day)),2) as avgtime_to_deliver,
FROM `orders` as O
INNER JOIN `customers` as C
ON O.customer_id=C.customer_id
GROUP BY C.customer_state),
B as (
SELECT customer_state as Highest_avg
FROM A
ORDER BY avgtime_to_deliver DESC
LIMIT 5),
C as (
SELECT customer_state as Lowest_avg
FROM A
ORDER BY avgtime_to_deliver ASC
LIMIT 5)
```

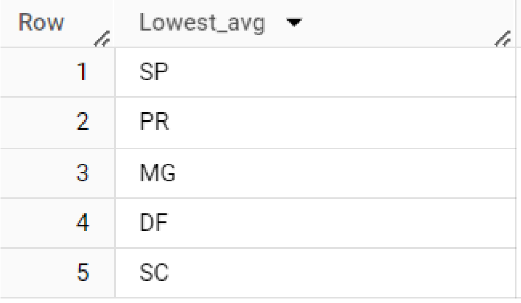
Output:

```
SELECT * FROM B
```



Row	Highest_avg
1	RR
2	AP
3	AM
4	AL
5	PA

```
SELECT * FROM C
```



Row	Lowest_avg
1	SP
2	PR
3	MG
4	DF
5	SC

Inference:

The States of Top 5 Highest and Lowest Average Delivery Time are computed. Similar to previous question, States with High average Delivery Time can have their Delivery Time reduced with interventions such as logistic vehicle routing and proper planning. Suitable Locations for warehousing can also be mapped to reduce Delivery Time. Incentives can also be provided to the employees who deliver faster to promote fast delivery.

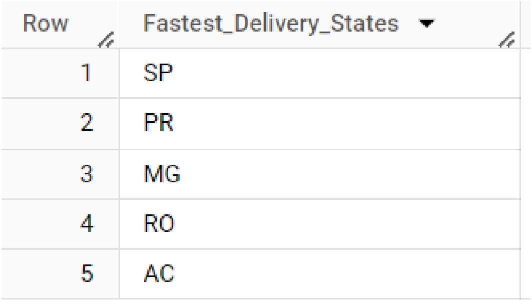
4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual &

estimated delivery date to figure out how fast the delivery was for each state.

Query:

```
WITH A as (
SELECT C.customer_state,
ROUND(AVG(DATE_DIFF(O.order_delivered_customer_date,O.order_purchase_timestamp,day)),2) as avgtime_to_deliver,
ROUND(AVG(DATE_DIFF(O.order_estimated_delivery_date,O.order_delivered_customer_date,day)),2) as avg_est_time_to_deliver
FROM `orders` as O
INNER JOIN `customers` as C
ON O.customer_id=C.customer_id
GROUP BY C.customer_state),
B AS
(SELECT *,ROUND(avg_est_time_to_deliver-avgtime_to_deliver,2) as `fastest`
FROM A
ORDER BY `fastest` DESC)
SELECT customer_state as `Fastest_Delivery_States`
from B
LIMIT 5;
```

Output:



Row	Fastest_Delivery_States
1	SP
2	PR
3	MG
4	RO
5	AC

Inference:

The Top 5 States where the order delivery is really fast as compared to the estimated date of delivery are computed. The performance of these states can be taken as benchmarks in identifying the parameters which are significantly responsible for faster delivery, and can be implemented in rest of the States to promote faster Delivery to improve the reputation of the Company.

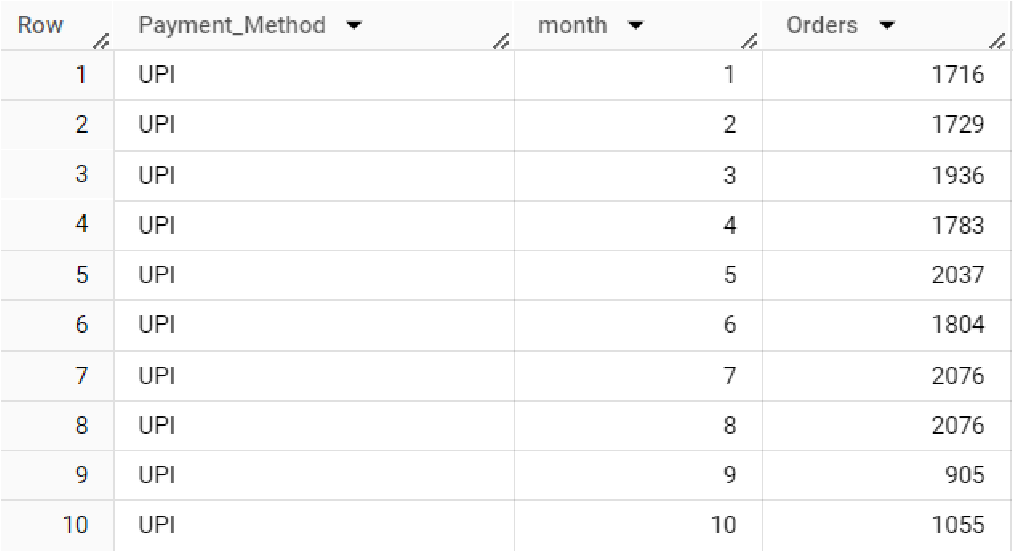
6. Analysis based on the payments:

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

Query:

```
SELECT P.payment_type as `Payment_Method`,extract(Month FROM
DATETIME(O.order_purchase_timestamp,"-3:00")) as `month`,
COUNT(O.order_id) as Orders
FROM `orders` as O
LEFT JOIN `payments` as P
ON O.order_id=P.order_id
WHERE payment_type is not null
GROUP BY P.payment_type, month
Order by P.payment_type, month asc;
```

Output:



Row	Payment_Method	month	Orders
1	UPI	1	1716
2	UPI	2	1729
3	UPI	3	1936
4	UPI	4	1783
5	UPI	5	2037
6	UPI	6	1804
7	UPI	7	2076
8	UPI	8	2076
9	UPI	9	905
10	UPI	10	1055

Inference:

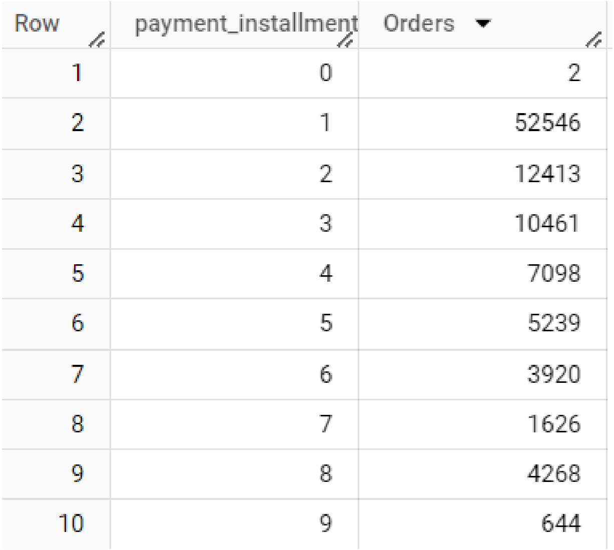
The most dominant form of Payment Method is Credit Card, followed by UPI. If company tends to have more profits in any specific Payment Method, Company can incentivise the Payment Method to maximize profit by rewarding the customer for choosing that specific Payment Method.

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

Query:

```
SELECT payment_installments, COUNT(order_id) as `Orders`
FROM `payments`
GROUP BY payment_installments
ORDER BY payment_installments ASC;
```

Output:



Row	payment_installment	Orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Inference:

The most dominant form of Payment Installment is one time payment, followed by 2 installments and 3 installments. This could denote that most of the customers either buy products of relatively lesser price or there are more customers from relatively high income groups. Comparing this with the types of Products purchased in this company could provide us with excellent insights.