SCALER ACADEMY

Target-SQL

A BUISNESS CASE STUDY

Soham Kavathekar

9/1/2023

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

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

1.1.1. Solution

```
SELECT
  column_name,
  data_type
FROM
  `target.INFORMATION_SCHEMA.COLUMNS`
WHERE
  table_name = 'customers'
```

1.1.2. Answer

Query results

SAVE RESULTS

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

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

1.1.3. Insight

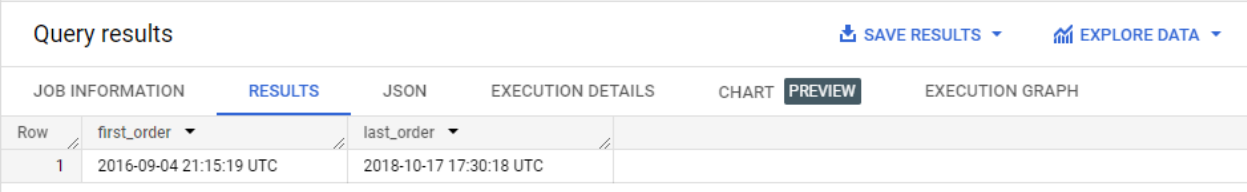
The Customer Table contains the customer details in 5 columns , detailing the Customer_ID, Customer_Unique_id which identifies the customer and their various purchases and next 3 columns provide the City, State and Zip code of the customer identifying the general location. This helps in identifying the zone wise business of products and the pre-loading the warehouse in particular area for particular items.

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

1.2.1. Solution

```
SELECT
min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order
FROM
target.orders;
```

1.2.2. Answer



1.2.3. Insight

This query provides the insight for the data acquisition duration.

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

1.3.1. Solution

```
WITH T2 as (SELECT
customer_city,
customer_state
FROM
`target.customers`
WHERE
customer_id IN (SELECT customer_id FROM `target.orders`)
GROUP BY customer_state, customer_city
ORDER BY customer_state,customer_city)

SELECT
count(DISTINCT customer_city) count_city,
count(DISTINCT customer_state) count_state
FROM T2
```

1.3.2. Answer

1.3.3. Insight

It gives information about the reach of the Target across the region. Comparing the current data with the region's number of state and number of cities will provide the region to be expanded in (city or state).

2. In-depth Exploration

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

2.1.1. Solution

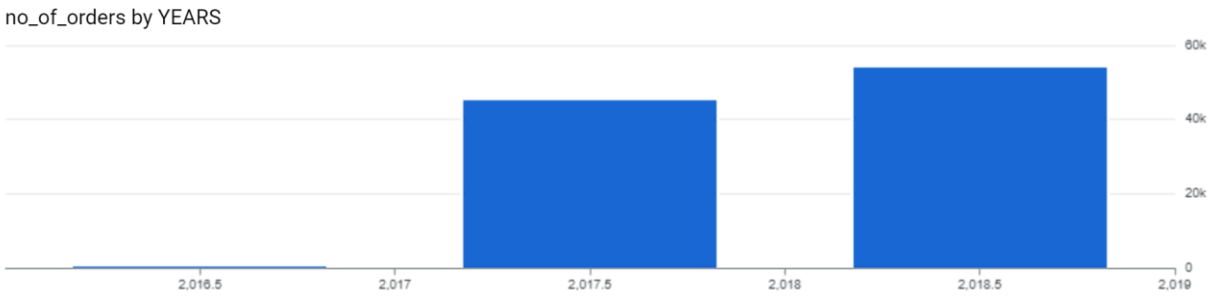
```
SELECT
*
FROM
(
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) as YEARS,
count(order_id) as no_of_orders
FROM
`target.orders`
GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp))
ORDER BY YEARS;
```

2.1.2. Answer

Query results

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

Row	YEARS	no_of_orders
1	2016	329
2	2017	45101
3	2018	54011



2.1.3. Insight

The YoY sale is in increasing, the YoY increase in no. of orders from 2017 to 2018 is 19.75%. This increase in no of orders does not indicate the Total Order amount increase YoY. The average order value is also a key factor.

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

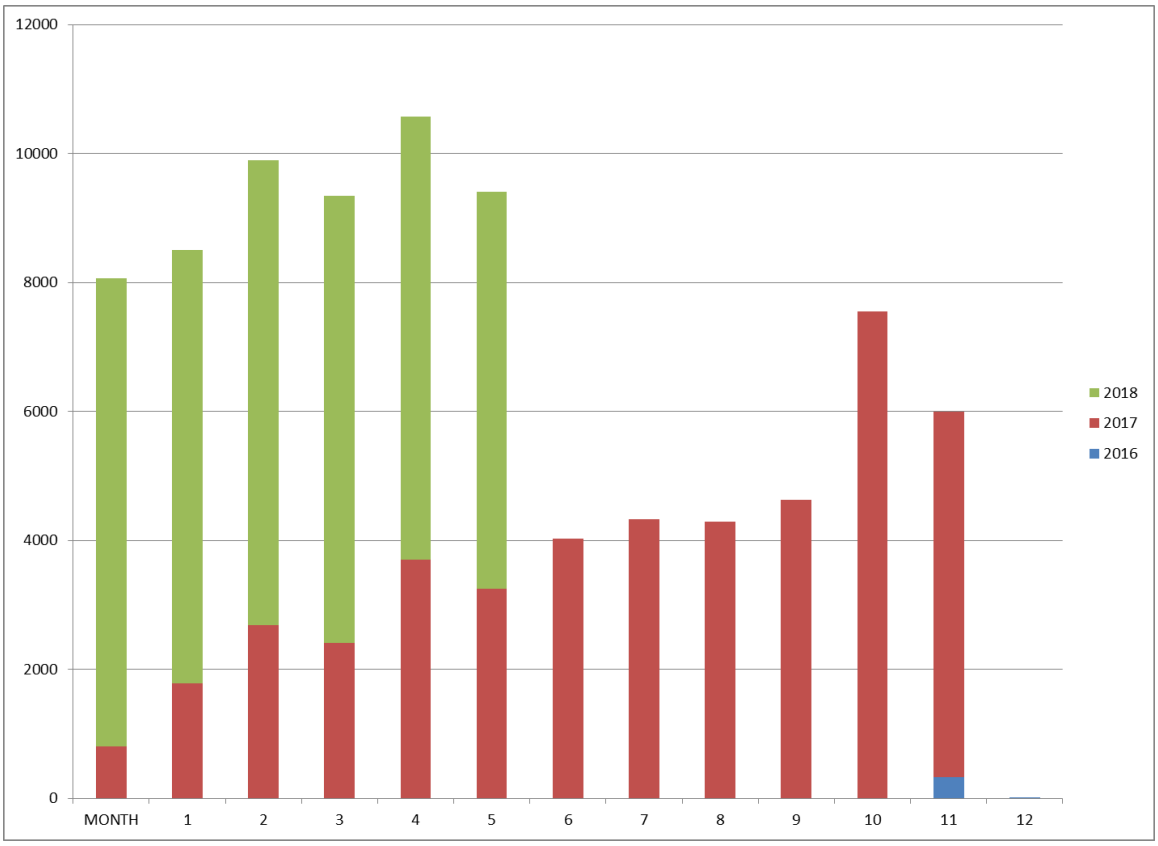
2.2.1. Solution

```
SELECT
*
FROM
(
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) as YEARS,
EXTRACT(MONTH FROM order_purchase_timestamp) as MNTH,
count(order_id) as no_of_orders
FROM
`target.orders`
GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp), EXTRACT(MONTH FROM
order_purchase_timestamp))
ORDER BY YEARS , MNTH;
```

2.2.2. Answer

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

Row	YEARS	MNTH	no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026



2.2.3. Insight

The MoM sales are increasing, highest no of sales were observed during the November 2017. Cumulative max sales were observed in the month of the April followed by February and May. The data is insufficient to provide deeper insight on the MoM sales over the years.

2.3. 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

2.3.1. Solution

```
WITH T2 as (SELECT
order_id,
CASE
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 0 and 6 THEN 'DAWN'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 7 and 12 THEN 'MORNING'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 13 and 18 THEN 'AFTERNOON'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 19 and 23 THEN 'NIGHT'
END as time_of_day
FROM
`target.orders`
ORDER BY order_id)
```

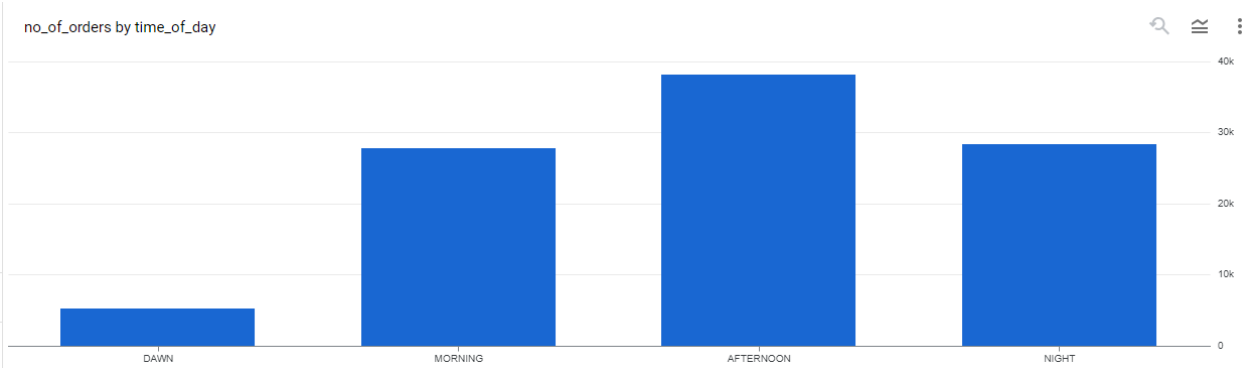
```sql
SELECT DISTINCT
time_of_day,
count(order_id) OVER(PARTITION BY time_of_day) as no_of_orders
FROM T2
ORDER BY
  CASE
    WHEN time_of_day = 'DAWN' THEN 1
    WHEN time_of_day = 'MORNING' THEN 2
    WHEN time_of_day = 'AFTERNOON' THEN 3
    WHEN time_of_day = 'NIGHT' THEN 4
  END
```

2.3.2. Answer

Query results

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

Row	time_of_day	no_of_orders
1	DAWN	5242
2	MORNING	27733
3	AFTERNOON	38135
4	NIGHT	28331



2.3.3. Insight

The orders are maximum in the Afternoon , followed by Morning and least favorite time to order for Brazilians is Dawn.

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

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

3.1.1. Solution

```
SELECT DISTINCT
EXTRACT(YEAR FROM order_purchase_timestamp) as YR,
EXTRACT(MONTH FROM order_purchase_timestamp) as MNTH,
c.customer_state as CS,
count(o.order_id) OVER (PARTITION BY EXTRACT(YEAR FROM order_purchase_timestamp),EXTRACT(MONTH
FROM order_purchase_timestamp),c.customer_state) as count_orders
FROM
`target.orders` o JOIN `target.customers` c ON o.customer_id = c.customer_id
ORDER BY CS, YR , MNTH
```

3.1.2. Answer

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

Row	CS	YR	MNTH	count_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3

*This data is difficult to put graphically currently.

3.1.3. Insight

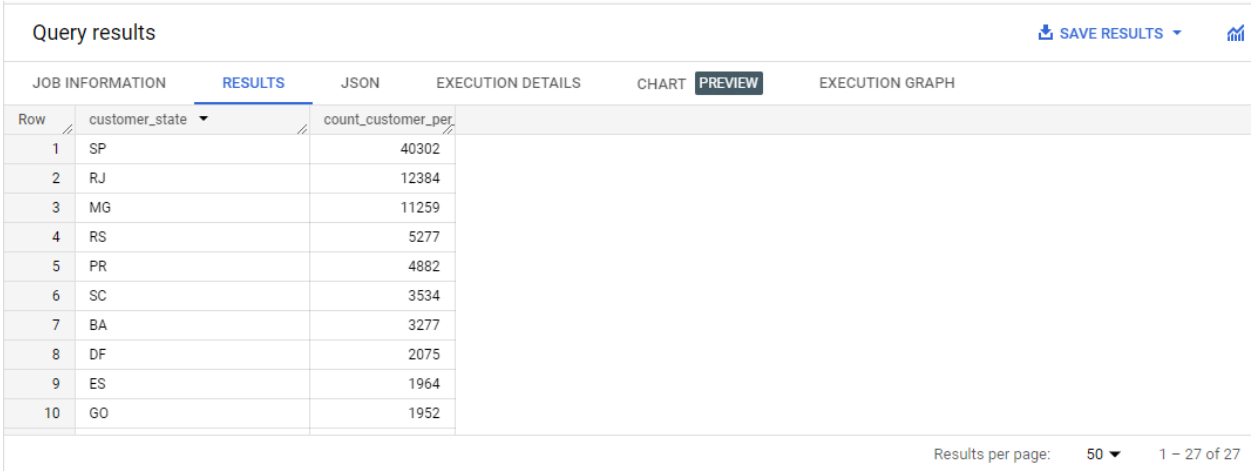
This query provides insight on the regional purchase rise and fall from customers providing insight on the native festivities and purchase requirements. This data also provides insight on the population density and economic demographic for purchases.

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

3.2.1. Solution

```
SELECT DISTINCT
customer_state,
count(customer_unique_id) OVER(PARTITION BY customer_state) AS count_customer_per_state
from
`target.customers`
ORDER BY count_customer_per_state DESC;
```

3.2.2. Answer



3.2.3. Insight

This provides data regarding customer concentration and allocation of marketing in low customer regions i.e. better customer acquisition.

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

4.1. 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.

4.1.1. Solution

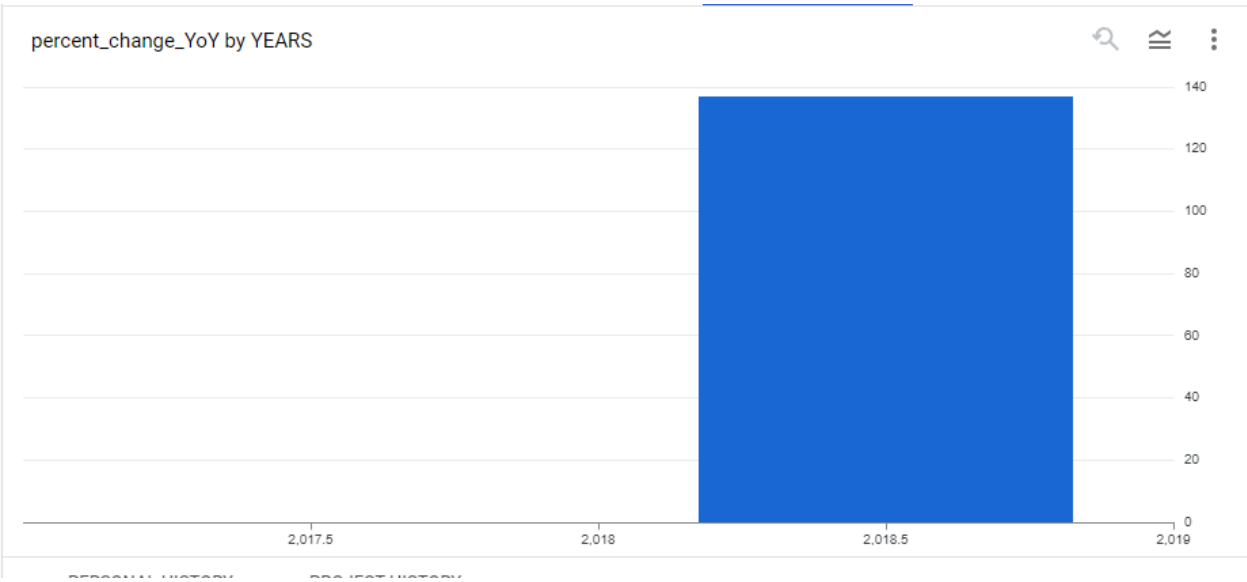
```
SELECT
*,
ROUND(ifNULL(((total_payments -lag(total_payments) OVER(ORDER BY YEARS
ASC))*100/lag(total_payments) OVER(ORDER BY YEARS ASC)), 0 ),2) as percent_change_YoY
FROM(
SELECT DISTINCT
EXTRACT(YEAR FROM order_purchase_timestamp) as YEARS,
sum(p.payment_value) OVER(PARTITION BY EXTRACT(YEAR FROM order_purchase_timestamp))
total_payments,
FROM
`target.payments` p JOIN `target.orders` o ON p.order_id = o.order_id
WHERE EXTRACT(MONTH FROM order_purchase_timestamp) between 1 and 8
ORDER BY YEARS)
ORDER BY YEARS;
```

4.1.2. Answer

Query results

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

Row	YEARS	total_payments	percent_change_YoY
1	2017	3669022.12	0.0
2	2018	8694733.84	136.98



4.1.3. Insight

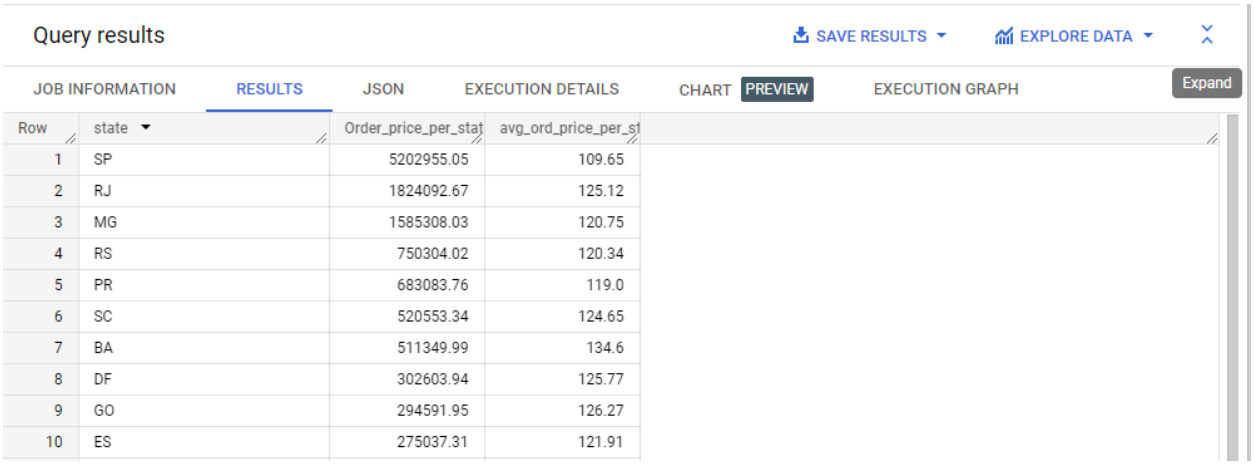
The graph shows YoY increase the total order values and there is 136.98 % increase in the cost of orders.

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

4.2.1. Solution

```
with cte as (
  SELECT o.order_id , c.customer_state
  FROM `target.orders` o JOIN `target.customers` c ON o.customer_id = c.customer_id
)

select DISTINCT
c.customer_state as state,
ROUND(sum(oi.price) OVER(PARTITION BY c.customer_state),2)as Order_price_per_state,
ROUND(sum(oi.price) OVER(PARTITION BY c.customer_state)/count(c.order_id) OVER (PARTITION BY
c.customer_state),2) as avg_ord_price_per_state
FROM cte c JOIN `target.order_items` oi ON c.order_id = oi.order_id
ORDER BY Order_price_per_state DESC;
```

4.2.2. Answer



Row	state	Order_price_per_stat	avg_ord_price_per_st
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

4.2.3. Insight

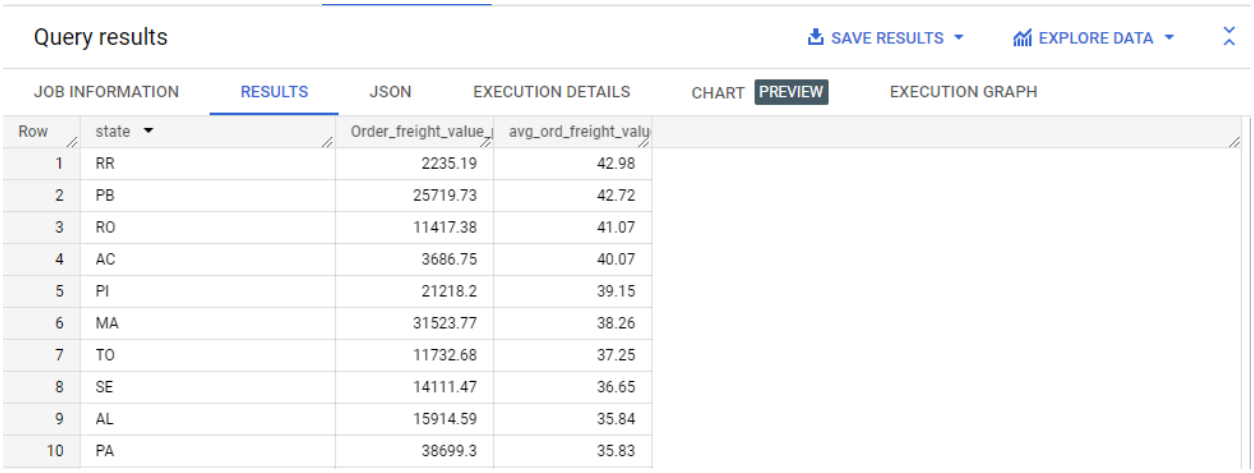
This data provides the order price by state which gives ideas about the warehouse locations to minimize the delivery time. Depending on the product types the stocking of products can be planned. The average order value tells about the frequency of use and need of use details.

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

4.3.1. Solution

```
with cte as (
  SELECT o.order_id , c.customer_state
  FROM `target.orders` o JOIN `target.customers` c ON o.customer_id = c.customer_id
)

select DISTINCT
c.customer_state as state,
ROUND(sum(oi.freight_value) OVER(PARTITION BY c.customer_state),2)as
Order_freight_value_per_state,
ROUND(sum(oi.freight_value) OVER(PARTITION BY c.customer_state)/count(c.order_id) OVER
(PARTITION BY c.customer_state),2) as avg_ord_freight_value_per_state
FROM cte c JOIN `target.order_items` oi ON c.order_id = oi.order_id
ORDER BY  avg_ord_freight_value_per_state DESC;
```

4.3.2. Answer



Row	state	Order_freight_value_	avg_ord_freight_valu
1	RR	2235.19	42.98
2	PB	25719.73	42.72
3	RO	11417.38	41.07
4	AC	3686.75	40.07
5	PI	21218.2	39.15
6	MA	31523.77	38.26
7	TO	11732.68	37.25
8	SE	14111.47	36.65
9	AL	15914.59	35.84
10	PA	38699.3	35.83

4.3.3. Insight

The average Freight value can be used as reference to plan cost effective shipping. The reduction in freight value will increase the profits.

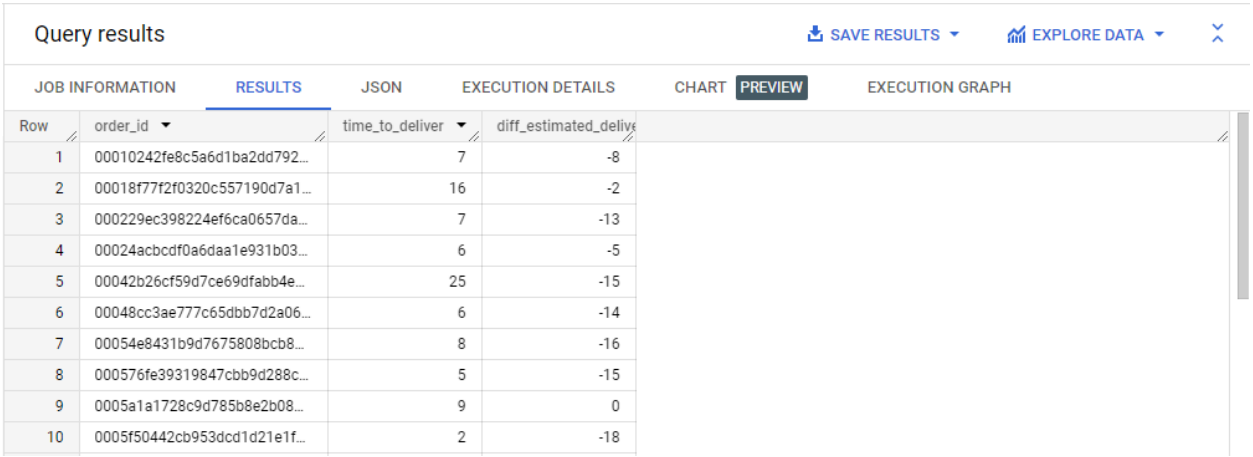
5. Analysis based on sales, freight and delivery time.

5.1. 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.

5.1.1. Solution

```
SELECT
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) as
time_to_deliver,
TIMESTAMP_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) as
diff_estimated_delivery
FROM `target.orders`
WHERE order_status = 'delivered'
ORDER BY order_id
```

5.1.2. Solution



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	time_to_deliver	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

5.1.3. Insight

This provides the insight on the supply chain / logistics prediction efficiency and areas of improvement in the supply chain. The negative value of ‘diff_estimated_delivery’ is showing orders delivered early and positive value shows delays in delivery in Days.

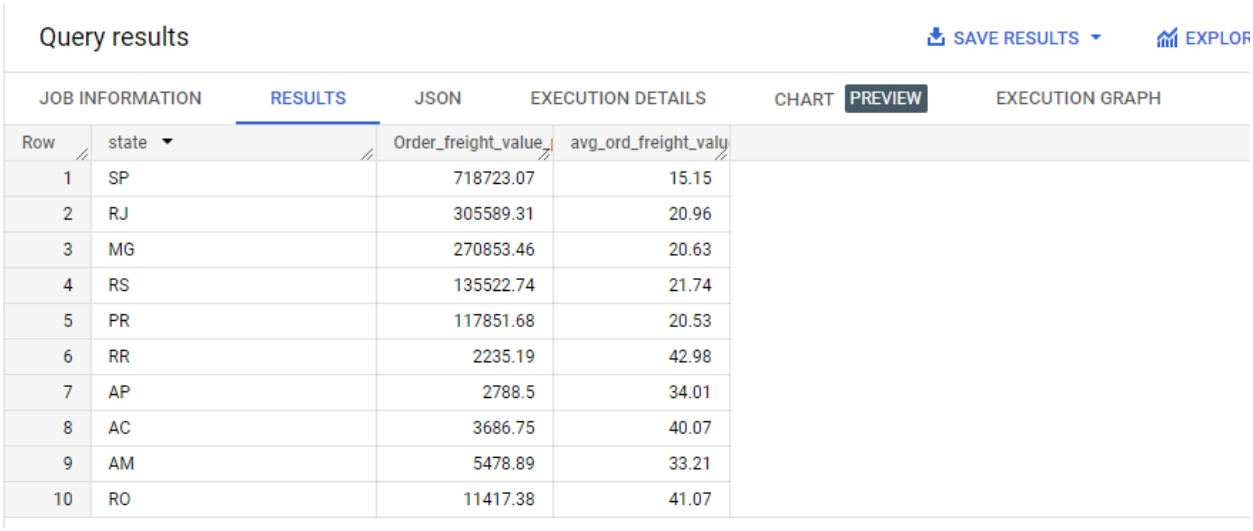
5.2. Find out the top 5 states with the highest & lowest average freight value.

5.2.1. Solution

```
with cte as (
  SELECT o.order_id , c.customer_state
  FROM `target.orders` o JOIN `target.customers` c ON o.customer_id = c.customer_id
)

(select DISTINCT
c.customer_state as state,
ROUND(sum(oi.freight_value) OVER(PARTITION BY c.customer_state),2)as
Order_freight_value_per_state,
ROUND(sum(oi.freight_value) OVER(PARTITION BY c.customer_state)/count(c.order_id) OVER
(PARTITION BY c.customer_state),2) as avg_ord_freight_value_per_state
FROM cte c JOIN `target.order_items` oi ON c.order_id = oi.order_id
ORDER BY Order_freight_value_per_state DESC
LIMIT 5)

UNION ALL

(SELECT DISTINCT
c.customer_state as state,
ROUND(sum(oi.freight_value) OVER(PARTITION BY c.customer_state),2)as
Order_freight_value_per_state,
ROUND(sum(oi.freight_value) OVER(PARTITION BY c.customer_state)/count(c.order_id) OVER
(PARTITION BY c.customer_state),2) as avg_ord_freight_value_per_state
FROM cte c JOIN `target.order_items` oi ON c.order_id = oi.order_id
```

```
ORDER BY Order_freight_value_per_state
LIMIT 5)
```

5.2.2. Answer



Query results

SAVE RESULTS | EXPLOR

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

Row	state	Order_freight_value_	avg_ord_freight_valu
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	RR	2235.19	42.98
7	AP	2788.5	34.01
8	AC	3686.75	40.07
9	AM	5478.89	33.21
10	RO	11417.38	41.07

5.2.3. Insight

The improvement is needed in the bottom 5 of the with respect to freight cost per order in those states. This means it is an area of improvement in the supply chain for better cost reduction.

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

5.3.1. Solution

```
WITH cte as
(
SELECT
order_id as orders,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) as
time_to_deliver,
c.customer_state as state
FROM `target.orders` o JOIN `target.customers` c ON o.customer_id = c.customer_id
WHERE order_status = 'delivered' AND (order_delivered_customer_date is not null and
order_purchase_timestamp is not null)
ORDER BY time_to_deliver DESC
)

(SELECT
state, ROUND(avg(time_to_deliver),1) as  AVERAGE_TIME, 'Least Days' as Type
FROM cte
GROUP BY state
ORDER BY AVERAGE_TIME ASC
LIMIT 5)
```

```
UNION ALL

(SELECT
state, ROUND(avg(time_to_deliver),1) as  AVERAGE_TIME, 'Most Days' as Type
FROM cte
GROUP BY state
ORDER BY AVERAGE_TIME DESC
LIMIT 5)
```

5.3.2. Answer

Query results

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

Row	state	AVERAGE_TIME	Type
1	RR	29.0	Most Days
2	AP	26.7	Most Days
3	AM	26.0	Most Days
4	AL	24.0	Most Days
5	PA	23.3	Most Days
6	SP	8.3	Least Days
7	PR	11.5	Least Days
8	MG	11.5	Least Days
9	DF	12.5	Least Days
10	SC	14.5	Least Days

5.3.3. Insight

This identifies regions with average delays state wise and average delay (+ve)/ prompt(0/+ve). The data shows needs in strengthening in the supply chain to reduce delays as there is no average value min/max zero or negative.

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

5.4.1. Solution

```
with cte as
(
SELECT
o.order_id as orders,
c.customer_state as state,
TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) as
estimate_vs_actual
FROM `target.orders` o JOIN `target.customers` c ON o.customer_id = c.customer_id
WHERE order_status = 'delivered'
ORDER BY order_id
```

```
)

select
state,
Round(-avg(estimate_vs_actual),2) as average_a_vs_e
from cte
group by state
ORDER BY average_a_vs_e DESC
LIMIT 5;
```

5.4.2. Answer

Query results

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

Row	state	average_a_vs_e
1	AL	-7.95
2	MA	-8.77
3	SE	-9.17
4	ES	-9.62
5	BA	-9.93

5.4.3. Insight

The data shows i.e. negative values , that the orders are getting delivered after the estimated date. The most close estimation date is ~8 days. This means requirement of supply chain strengthening.

6. Analysis based on the payments

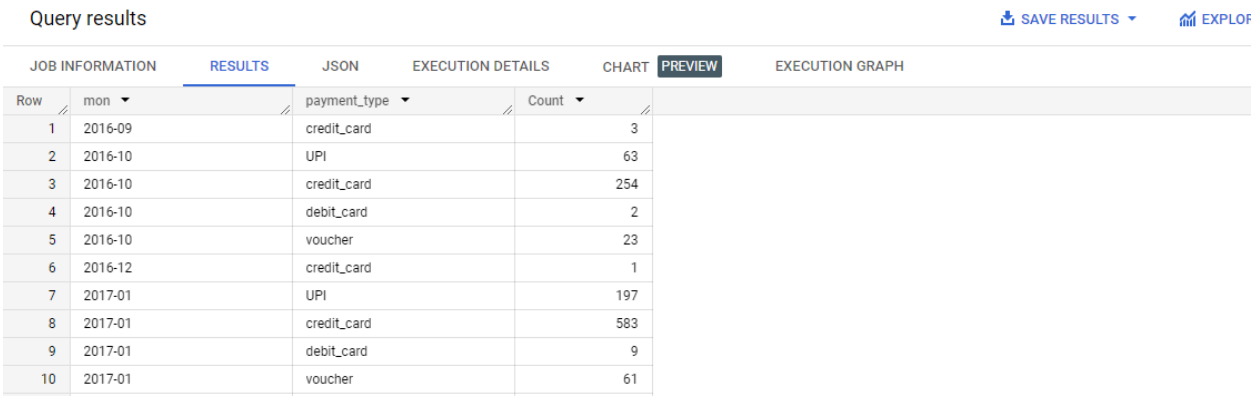
6.1. Find the month on month no. of orders placed using different payment types.

6.1.1. Solution

```
with cte as
(
select
FORMAT_DATE('%Y-%m', o.order_purchase_timestamp) as mon,payment_type, count(p.payment_type) as
Count
from `target.orders` o JOIN  `target.payments` p ON o.order_id = p.order_id
GROUP BY FORMAT_DATE('%Y-%m', o.order_purchase_timestamp), payment_type
)

select
*
FROM cte
```

```
ORDER BY mon,payment_type;
```

6.1.2. Answer



Query results

SAVE RESULTS | EXPLOR

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

Row	mon	payment_type	Count
1	2016-09	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	2017-01	UPI	197
8	2017-01	credit_card	583
9	2017-01	debit_card	9
10	2017-01	voucher	61

6.1.3. Insight

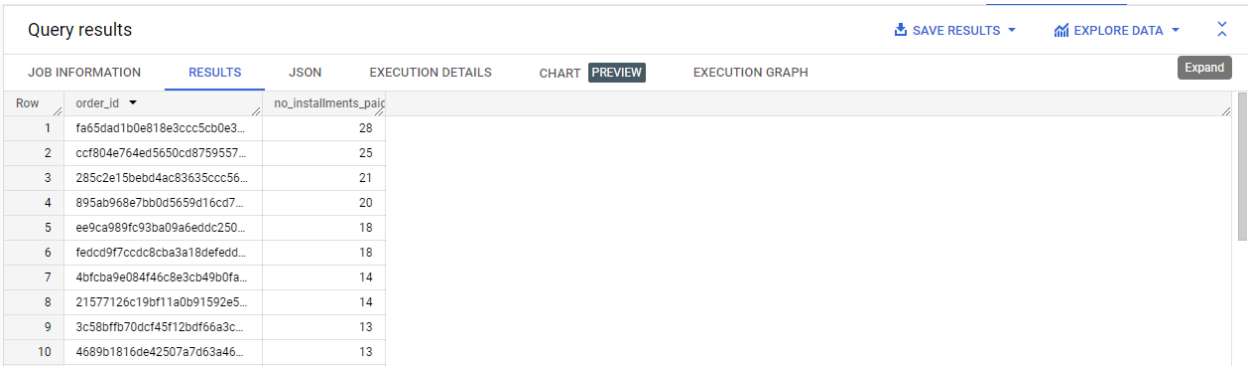
There is increase in UPI and credit card payments MoM. If there is need to be more debit card and other delivery payment , those payments methods to be incentivized.

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

6.2.1. Solution

```
with cte as
(
select
order_id ,
count(payment_installments) as no_installments_paid
FROM `target.payments`
WHERE payment_sequential <> 1
GROUP BY order_id
)

select
*
from cte
order by cte.no_installments_paid DESC
```

6.2.2. Answer



Row	order_id	no_installments_paid
1	fa65dad1b0e818e3ccc5cb0e3...	28
2	ccf804e764ed5650cd8759557...	25
3	285c2e15bebd4ac83635ccc56...	21
4	895ab968e7bb0d5659d16cd7...	20
5	ee9ca989fc93ba09a6eddc250...	18
6	fedcd9f7ccdc8cba3a18defedd...	18
7	4bfcba9e084f46c8e3cb49b0fa...	14
8	21577126c19bf11a0b91592e5...	14
9	3c58bffb70dcf45f12bdf66a3c...	13
10	4689b1816de42507a7d63a46...	13

6.2.3. Insight

The EMI’s which are paid indicates that the customer is diligent and pays installments on-time. The customer’s credit limit can be increased.