PROJECT SQL

Q1) Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1) The columns and their data type in the customer table are as follows

Column Name	Data Type
customer_id	String
customer_unique_id	String
customer_city	String
customer_state	String
customer_zip_code_prefix,type	Integer

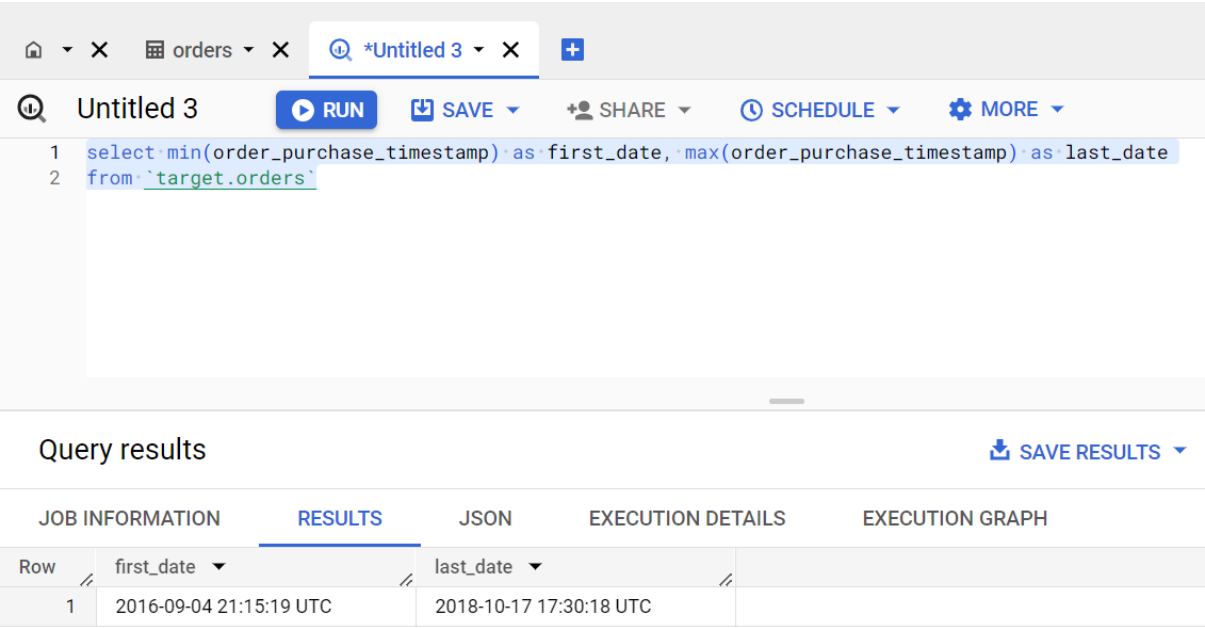
2) Get the time range between which the orders were placed.

The query for knowing the range of date within which the orders were placed is

"select min(order_purchase_timestamp) as first_date, max(order_purchase_timestamp) as last_date

from `company.orders`"

The min() function returned the minimum date since the orders were placed and the max() returned the last date when the orders were placed.



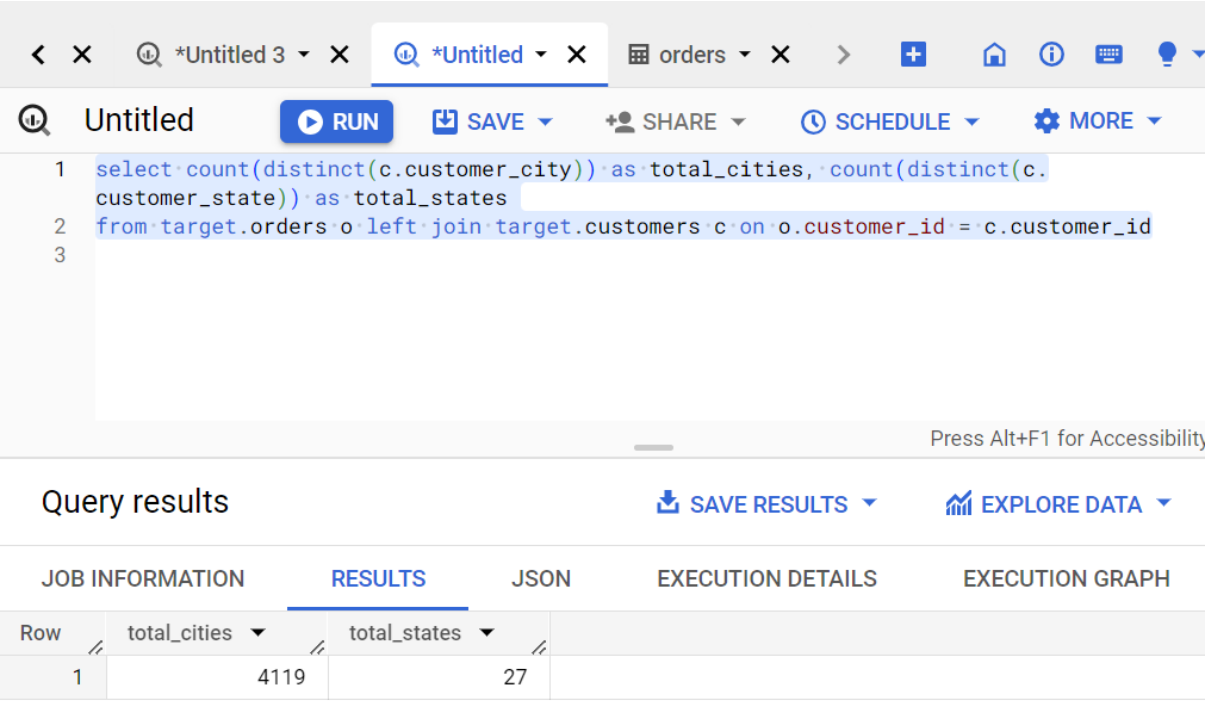
So the orders were placed from 4th September 2016 9:15:19 PM to 17th October 2018 5:30:18 PM both timestamps inclusive

3) 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 company.orders o left join company.customers c on o.customer_id = c.customer_id"

I used distinct count so that we must know the count of different cities and states from where our customers are ordering which will help us to understand our actual reach in the market. As the table used is 'orders' the timestamp is itself in the desired range of minimum and maximum date so we need not apply any filter for the same. The output received is as follows:



Q2) In-depth Exploration:

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

   Yes there is a growing pattern in the numbers of orders placed across the years with minimum being in 2016 where the total orders placed were just 329 but there was a boom in 2017 where the orders placed increased exponentially with whooping 45101, that's an increase of around 13608.5% which is fantastic, orders placed in 2017 and the trend held itself in 2018 too with orders increased to 54011 that amounts to increase in 19.75%

The query that helped me to derive the results is :

“select extract(year FROM (order_purchase_timestamp)) as year, count(order_id) as total_orders

from `company.orders`
group by 1
order by 1”

```
select extract(year FROM (order_purchase_timestamp)) as year, count(order_id) as
total_orders
from `target.orders`
group by 1
order by 1
```

Press Alt+F1 for Accessibility

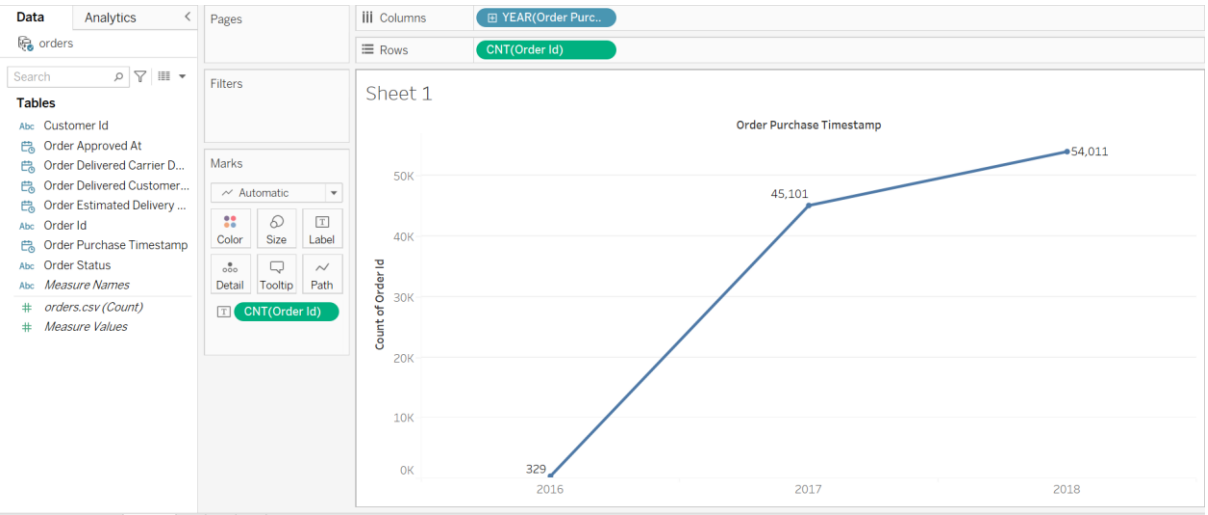
Query results

SAVE RESULTS EXPLORE DATA

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

Row	year	total_orders
1	2016	329
2	2017	45101
3	2018	54011

The trend can also be seen with the help of a line plot given below



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

The query to find the asked detail is:
select extract(month FROM (order_purchase_timestamp)) as month, count(order_id) as total_orders

from `company.orders`
group by 1
order by 1

The output received is pasted below

customers | *Untitled 3 | *Untitled | *Untitled 2 | products

Untitled 2 RUN SAVE SHARE SCHEDULE MORE

```
select extract(month FROM (order_purchase_timestamp)) as month, count(order_id) as total_orders
from `target.orders`
group by 1
order by 1
```

Press Alt+F1 f

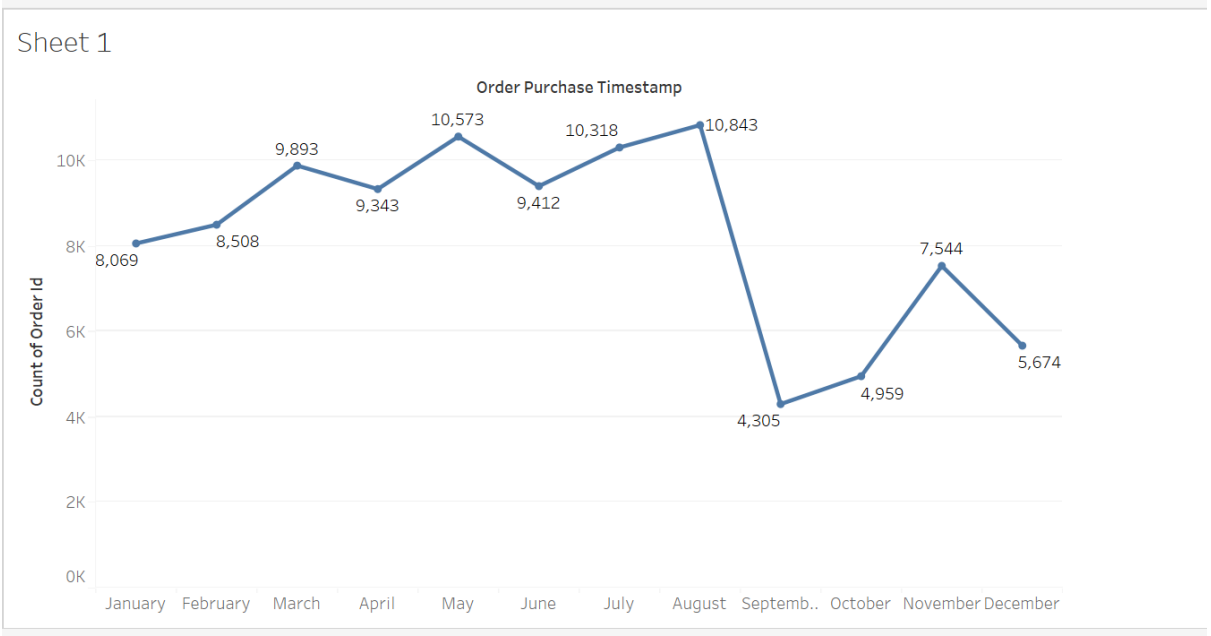
Query results SAVE RESULTS EXPLOR

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

Row	month	total_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412

The trend can also be seen by the line plot

Columns: MONTH(Order Pu..
Rows: CNT(Order Id)

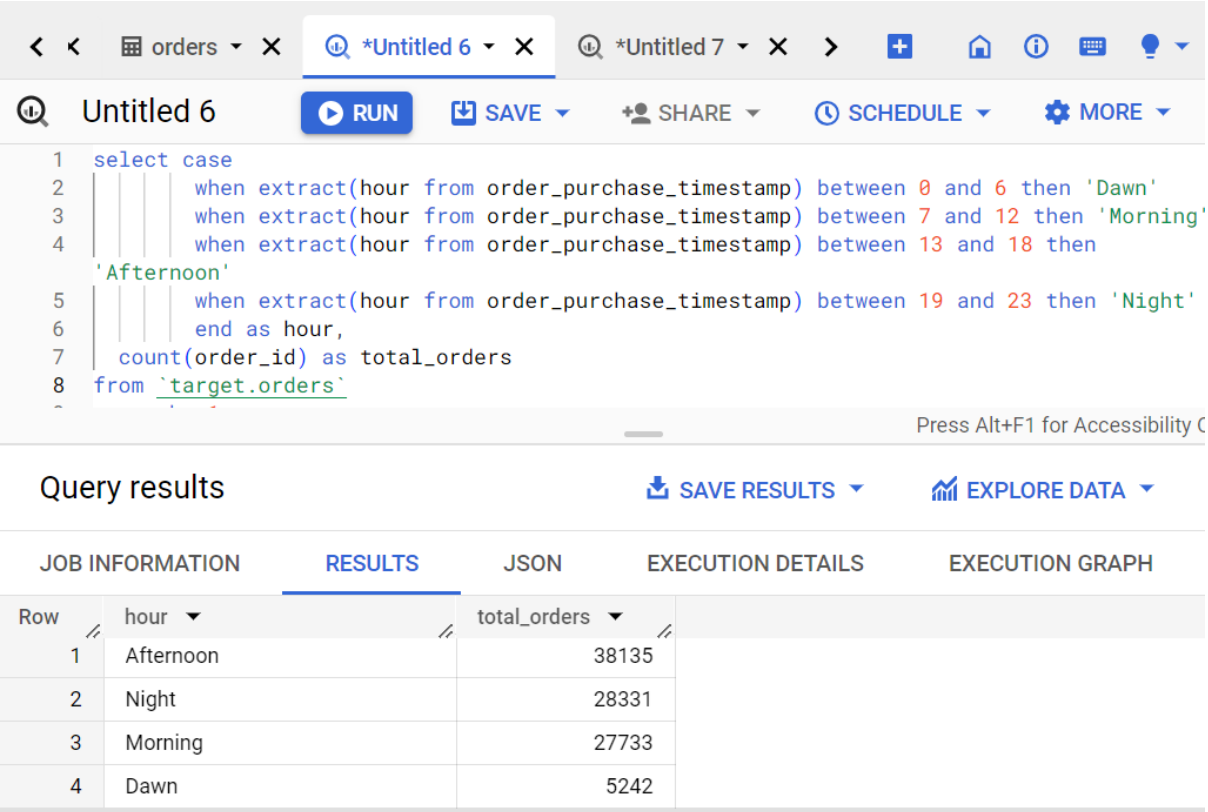


3) During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

The used query is :
"select case

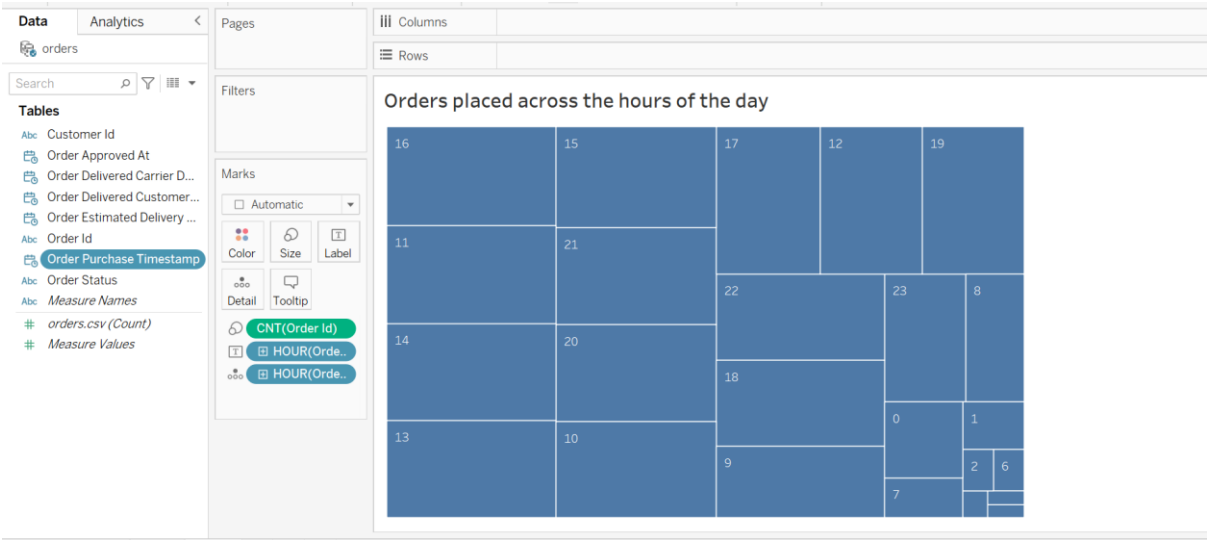
when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Morning'
when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
when extract(hour from order_purchase_timestamp) between 19 and 23 then 'Night'
end as hour,
count(order_id) as total_orders
from `company.orders`
group by 1
order by 2 desc"

OUTPUT:



Row	hour	total_orders
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

It can easily be concluded that Brazilians order maximum during afternoon that is between 13 hours to 18 hours and with the help of visualisation I can claim that the hour that receives the maximum order is 16:00 hour with 13:00 hour and 14:00 hour also in the proximity with their numbers so maybe the citizens shop when they find themselves free to shop online or use their free time for shopping while at work or after lunch. We can give out our best offers at that time to increase the sale further.



The digits 16,11,14,13 represent the time of the day

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

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

The query to find m-o-m sale of each state is :

"select month,state, orders, lead(orders) over(partition by state order by month) as m_o_m

from
(select count(order_id) as orders,c.customer_state as state, extract(month from order_purchase_timestamp) as month
from `company.orders` o left join `company.customers` c on o.customer_id = c.customer_id
group by 2,3) as a"

The output received is:

*Untitled | orders | *Untitled 2 | customers

Untitled 2 RUN SAVE SHARE SCHEDULE MORE

```
select month,state, orders, lead(orders) over(partition by state order by month) as m_o_m
from
(select count(order_id) as orders,c.customer_state as state, extract(month from order_purchase_timestamp) as month
from `target.orders` o left join `target.customers` c on o.customer_id = c.customer_id
group by 2,3) as a
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

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

Row	month	state	orders	m_o_m
1	1	MA	66	67
2	2	MA	67	77
3	3	MA	77	73
4	4	MA	73	65

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

The query to find the distribution of customers across the state is:
" select customer_state, count(distinct(customer_id)) as total_cutomers

from `company.customers`
group by 1"

The output received is given below:

*Untitled 2 | customers | *Untitled 3

Untitled 3 RUN SAVE SHARE SCHEDULE MORE

```
select customer_state, count(distinct(customer_id)) as total_cutomers
from `target.customers`
group by 1
```

Press Alt-

Query results SAVE RESULTS EXP

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

Row	customer_state	total_cutomers
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637

The least number of customers are from the state RN so we can give out some offers to the people there. The low customer count could also indicate to the low connectivity there and less awareness about the ease of shopping online.

Q4) **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).

The query is given below:

```
"select year, month, amount_paid,lead(amount_paid) over(partition by year order by month) as next_month_sale, ((lead(amount_paid) over(partition by year order by month)-amount_paid)/(amount_paid))*100 as growth

from
(select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) as month,sum(p.payment_value) as amount_paid
from `company.orders` o left join `company.payments` p on o.order_id = p.order_id
where extract(year from order_purchase_timestamp) between 2017 and 2018
group by 1,2) as a
where month between 1 and 8
order by 1,2"
```

The output is given below:

*Untitled 2 | customers | *Untitled 3 | *Untitled | payments | *Untitled 5

Untitled 5 RUN SAVE SHARE SCHEDULE MORE

```
select year, month, amount_paid,lead(amount_paid) over(partition by year order by month) as next_month_sale, ((lead(amount_paid) over(partition
by year order by month)-amount_paid)/(amount_paid))*100 as growth
from
(select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) as month,sum(p.payment_value) as
amount_paid
from `target.orders` o left join `target.payments` p on o.order_id = p.order_id
where extract(year from order_purchase_timestamp) between 2017 and 2018
group by 1,2) as a
```

Press Alt+F1 for Accessibility O

Query results SAVE RESULTS EXPLORE DATA

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

Row	year	month	amount_paid	next_month_sale	growth
1	2017	1	138488.0399999...	291908.0099999...	110.7821079712...
2	2017	2	291908.0099999...	449863.6000000...	54.11142708965...
3	2017	3	449863.6000000...	417788.0300000...	-7.13006564656...
4	2017	4	417788.0300000...	592918.8200000...	41.91857531198...
5	2017	5	592918.8200000...	511276.3800000...	-13.7695814749...
6	2017	6	511276.3800000...	592382.9200000...	15.86354135898...

Results per page: 50 1 – 16 of 16

The %growth started with a high value in the year 2017 and later got cooled down to fairly low level which indicates downward trend in the consumption pattern of people indicating low growth in the economy as well. The second quarter of the economy looked even more bleak than the first one.

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

"select c.customer_state, sum(p.payment_value) as total_sum , avg(p.payment_value) as average
from `company.orders` o left join `company.customers` c on o.customer_id = c.customer_id left join `company.payments`p on o.order_id = p.order_id
group by 1
order by 2"

Output is given below:

customers | *Untitled 3

Untitled 3 RUN SAVE SHARE SCHEDULE MORE

```
select c.customer_state, sum(p.payment_value) as total_sum , avg(p.payment_value) as average
from `target.orders` o left join `target.customers` c on o.customer_id = c.customer_id left join `target.payments`p on o.order_id = p.order_id
group by 1
order by 2
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

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

Row	customer_state	total_sum	average
1	RR	10064.62	218.7960869565...
2	AP	16262.79999999...	232.3257142857...
3	AC	19680.61999999...	234.2930952380...
4	AM	27966.93	181.6034415584...

Results per page: 50 1 – 27 of 27

PERSONAL HISTORY PROJECT HISTORY REFRESH

The least order value is received from the state RR which is also because the freight charges are very high and customers definitely don't like to pay from their pockets for the delivery. So to increase the sale from the state we must work on reducing freight charges for the state.

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

"

select c.customer_state, sum(oi.freight_value) as total_freight, avg(oi.freight_value) as average_freight

from `company.orders` o left join `company.customers` c on o.customer_id = c.customer_id
left join `company.order_items` oi on oi.order_id = o.order_id
group by 1
order by 2
"

output is given below

*Untitled12 *Untitled13 *Untitled14 *Untitled15

Untitled 15 RUN SAVE SHARE SCHEDULE MORE

```
select  c.customer_state, sum(oi.freight_value) as total_freight, avg(oi.freight_value) as average_freight

from `target.orders` o left join `target.customers` c on o.customer_id = c.customer_id
left join `target.order_items` oi on oi.order_id = o.order_id
group by 1
order by 2
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	total_freight	average_freight
1	RR	2235.19	42.98442307692...
2	AP	2788.500000000...	34.00609756097...
3	AC	3686.749999999...	40.07336956521...
4	AM	5478.889999999...	33.20539393939...
5	RO	11417.37999999...	41.06971223021...
6	TO	11732.68000000...	37.24660317460...
7	SE	14111.46999999...	36.65316883116...

Results per page: 50 1 – 27 of 27

The total freight is maximum for the state RR and the average freight is also high for it. We should look into it to reduce the charges which will help us increase our profit also reduce the cost of delivery for us and customers too.

Q5) 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.

"
select order_id,ceiling(((extract(hour from order_delivered_customer_date)) - extract(hour from order_purchase_timestamp))/24) as time_to_deliver, ceiling((extract(hour from order_delivered_customer_date) - extract(hour from order_estimated_delivery_date))/24) as diff_estimated_delivery
from `company.orders`
order by 2 desc
"

OUTPUT IS GIVEN BELOW

payments | *Untitled12 | *Untitled13 | *Untitled14

Untitled 14 RUN SAVE SHARE SCHEDULE MORE

```
select order_id,ceiling(((extract(hour from order_delivered_customer_date)) - extract( hour from
order_purchase_timestamp ))/24) as time_to_deliver, ceiling((extract(hour from order_delivered_customer_date)
- extract(hour from order_estimated_delivery_date))/24) as diff_estimated_delivery
from `target.orders`
order by 2 desc
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

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

Row	order_id	time_to_deliver	diff_estimated_deliv
1	770d331c84e5b214bd9dc70a1...	1.0	1.0
2	1950d777989f6a877539f5379...	1.0	1.0
3	dabf2b0e35b423f94618bf965f...	1.0	1.0
4	a8aa2cd070eeac7e4368cae3d...	1.0	1.0
5	813c55ce9b6baa8f879e064fbf...	1.0	1.0
6	ddf16d77e858a32f36e10c289...	1.0	1.0
7	4c3812e7963e6b0932c6142e3...	1.0	1.0
8	ddcf6c758c04fe3f2e0d1ae301...	1.0	1.0

We can clearly see that our delivery department is doing a great job as the maximum time it has taken us to deliver an order is 1 day and our estimated delivery date is nicely met. Several orders have been delivered the same day as the time taken to deliver them was 0.

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

Query for low_freight states

"select c.customer_state, row_number()over(order by(avg(p.freight_value)) asc) as low_freight_state
from `company.orders` o left join `company.customers` c on o.customer_id = c.customer_id left join `company.order_items` p on o.order_id = p.order_id
group by 1
limit 5"

The output is below:

The lowest average cost of freight is for state SP, PR, MG, RJ, DF in that order

*Untitled 3 | *Untitled 4 | customers | order

Untitled 3 RUN SAVE SHARE SCHEDULE MORE

```
select c.customer_state, row_number()over(order by(avg(p.freight_value)) asc) as low_freight_state
from `target.orders` o left join `target.customers` c on o.customer_id = c.customer_id left join `target.
order_items` p on o.order_id = p.order_id
group by 1
limit 5
```

Press Alt+F1 for Accessibility

Query results SAVE RESULTS EXPLORE DATA

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

Row	customer_state	low_freight_state
1	SP	1
2	PR	2
3	MG	3
4	RJ	4
5	DF	5

Query for high freight states:

"select c.customer_state, row_number()over(order by(avg(p.freight_value)) desc) as high_freight_state
from `company.orders` o left join `company.customers` c on o.customer_id = c.customer_id left join `company.order_items` p on o.order_id = p.order_id
group by 1
limit 5"

The output is below:

*Untitled 3 | *Untitled 4 | customers | order

Untitled 3 RUN SAVE SHARE SCHEDULE MORE

```
select c.customer_state, row_number()over(order by(avg(p.freight_value)) desc) as high_freight_state
from `target.orders` o left join `target.customers` c on o.customer_id = c.customer_id left join `target.
order_items` p on o.order_id = p.order_id
group by 1
limit 5
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

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

Row	customer_state	high_freight_state
1	RR	1
2	PB	2
3	RO	3
4	AC	4
5	PI	5

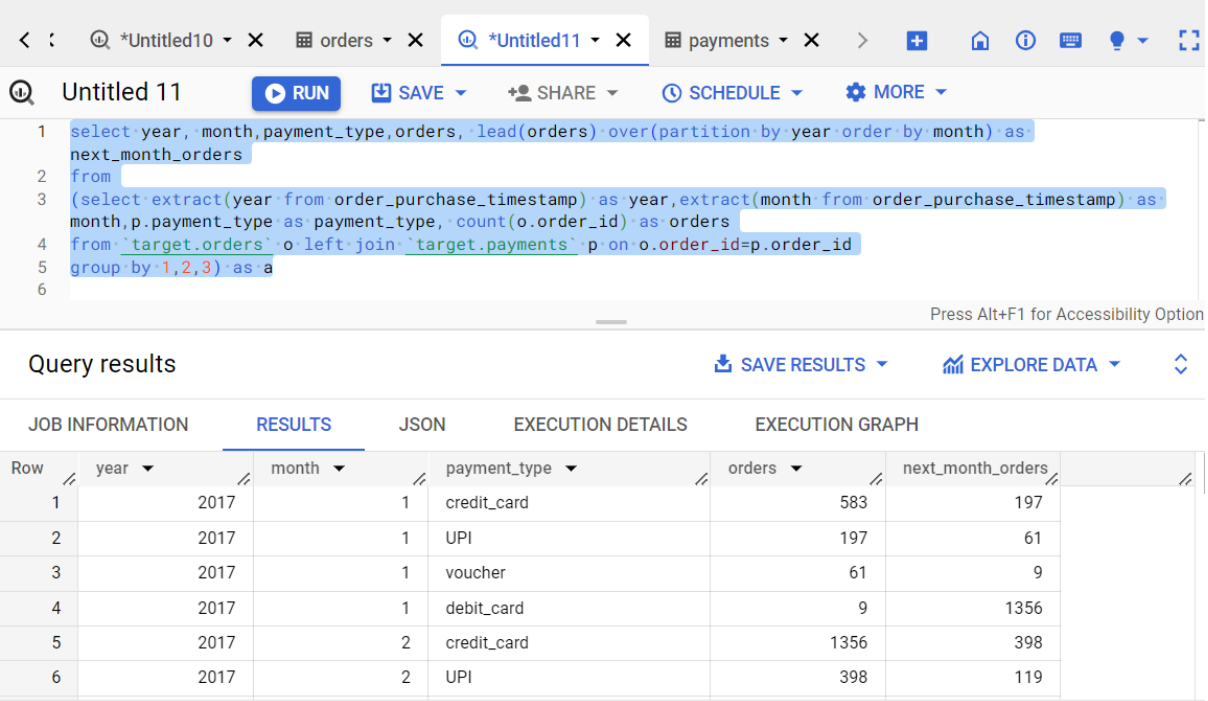
The highest average freight is for states RR,PB,RO,AC,PI in that order

Q6) Analysis based on the payments

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

" select year, month,payment_type,orders, lead(orders) over(partition by year order by month) as next_month_orders
from
(select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) as month,p.payment_type as payment_type, count(o.order_id) as orders
from `company.orders` o left join `company.payments` p on o.order_id=p.order_id
group by 1,2,3) as a"

output is given below:



Row	year	month	payment_type	orders	next_month_orders
1	2017	1	credit_card	583	197
2	2017	1	UPI	197	61
3	2017	1	voucher	61	9
4	2017	1	debit_card	9	1356
5	2017	2	credit_card	1356	398
6	2017	2	UPI	398	119

It can be inferred that the most favoured mode of payment is credit card as the maximum orders have been placed by using them, UPI follows it as the second highest mode of payment. The least favoured one is debit card . The company can rope in maximum number of credit card service providers to boost the sale further. It can also give some offer to credit card users so as to increase the ordering amount and quantity. To increase the sale from debit card we can give an option to save cards for faster checkouts and also give some offers or cashbacks on them.

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

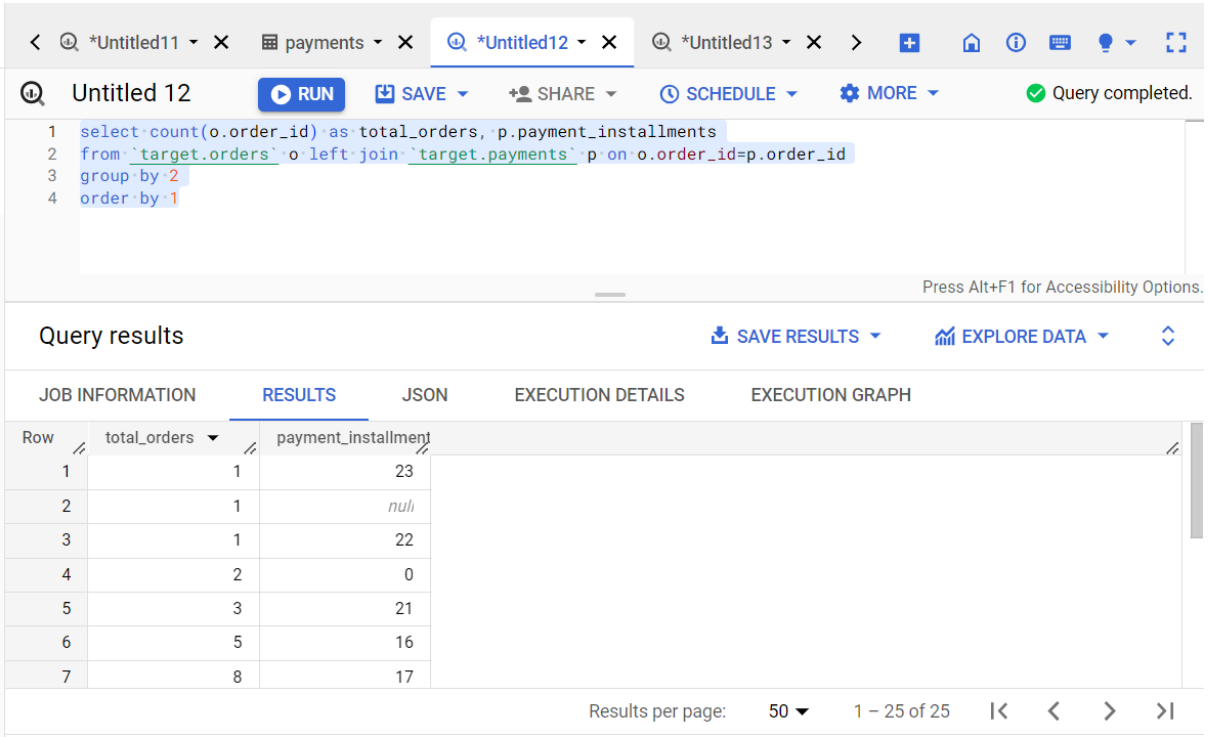
The query is:

“

select count(o.order_id) as total_orders, p.payment_installments
from `company.orders` o left join `company.payments` p on o.order_id=p.order_id
group by 2
order by 1

”

The output is given below:



It can be inferred from the data that payment instalments of short durations are most preferred with the most favourite ones being 1,2,3. Below is the output to support my claim

*Untitled11 | payments | *Untitled12 | *Untitled13

Untitled 12 RUN SAVE SHARE SCHEDULE MORE Query completed.

```
select count(o.order_id) as total_orders, p.payment_installments
from `target.orders` o left join `target.payments` p on o.order_id=p.order_id
group by 2
order by 1 desc
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

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

Row	total_orders	payment_installment
1	52546	1
2	12413	2
3	10461	3
4	7098	4
5	5328	10
6	5239	5
7	4268	8

Results per page: 50 1 – 25 of 25

The trend is because maximum number of purchases are made from credit cards and so people try to avoid interest payments or longer duration of payments.