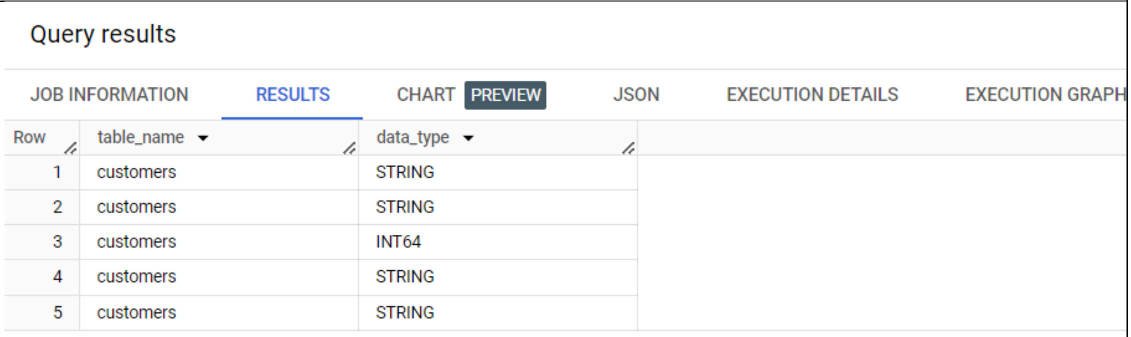
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

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

```
select table_name,data_type
from `businesscase1-403903.windowdate.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers';
```

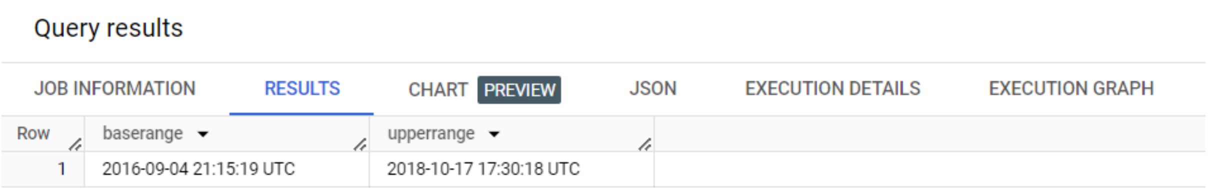


Row	table_name	data_type
1	customers	STRING
2	customers	STRING
3	customers	INT64
4	customers	STRING
5	customers	STRING

Insights: 20 percent of data is integer and 80 percent is string. Storage of integer is different from storage of string in database.

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

```
select min(order_purchase_timestamp) as baserange, max(order_purchase_timestamp) as
upperrange
from `windowdate.orders`
```

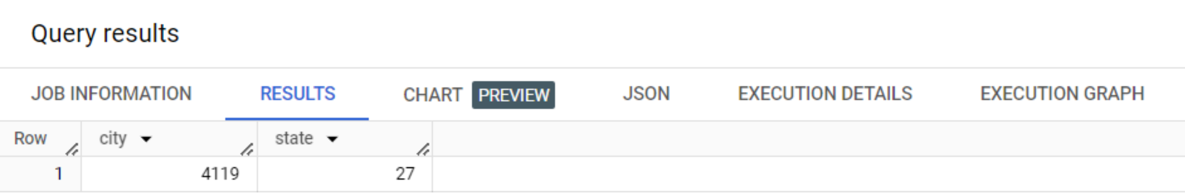


Row	baserange	upperrange
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insight : The timerange varies over almost a month over a year from 2016 to 2017. The months include September,2016 to October,2017.

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

```
select count(distinct customer_city) as city,count(distinct customer_state) as
state
from `windowdate.customers`
```



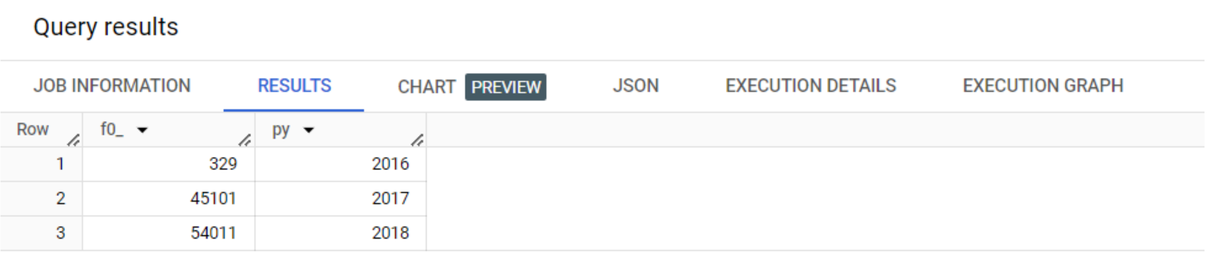
Row	city	state
1	4119	27

Insight : The records contains 27 states and approx. 4100 cities. The sates are less in number so any analysis would be easy to conclude if done on the basis of state. Macro Analysis could be done on the basis of state and then if needed we can analyse city , state wise for any micro analysis.

2. **In-depth Exploration:**

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

```
select count(*),
extract(year from `order_purchase_timestamp`) as py
from `windowdate.orders`
group by py
order by py asc;
```

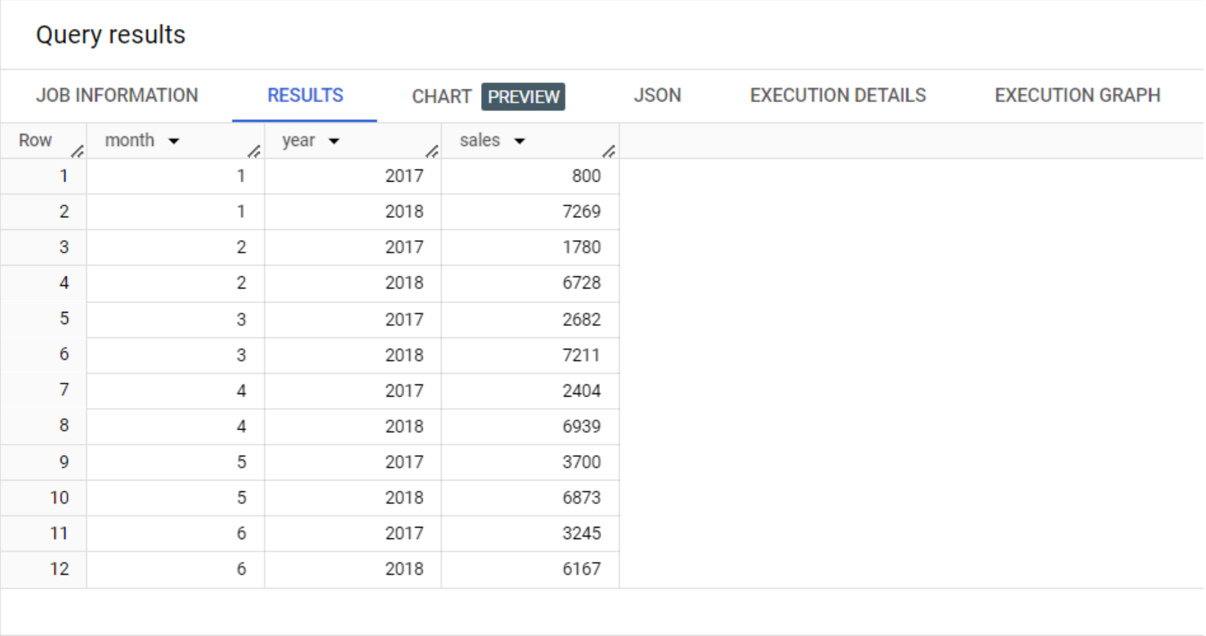


Row	f0_	py
1	329	2016
2	45101	2017
3	54011	2018

Yes, the trend has exponentially increased from 2016 to 2017 and linearly from 2017 and 2018.

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

```
select extract(month from `order_purchase_timestamp`) as month,
extract(year from `order_purchase_timestamp`) as year,
count(*) as sales
from `windowdate.orders`
group by month, year
order by month, year;
```



Query results

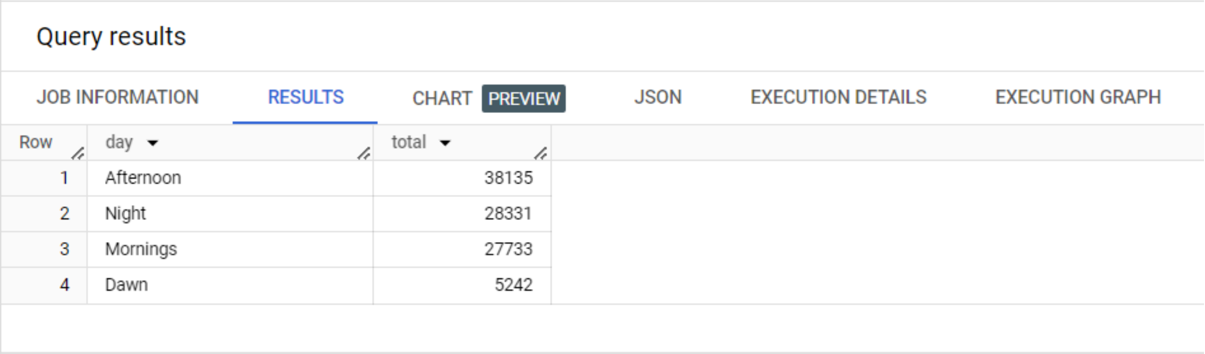
Row	month	year	sales
1	1	2017	800
2	1	2018	7269
3	2	2017	1780
4	2	2018	6728
5	3	2017	2682
6	3	2018	7211
7	4	2017	2404
8	4	2018	6939
9	5	2017	3700
10	5	2018	6873
11	6	2017	3245
12	6	2018	6167

During February and March the the sales almost trippled from 2017 to 2018. During May and June the the sales almost doubled from 2017 to 2018.

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

```
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
'Mornings'
   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 day,
   count(*) as total
 from `windowdate.orders`
 group by day
 order by total desc
```



Query results

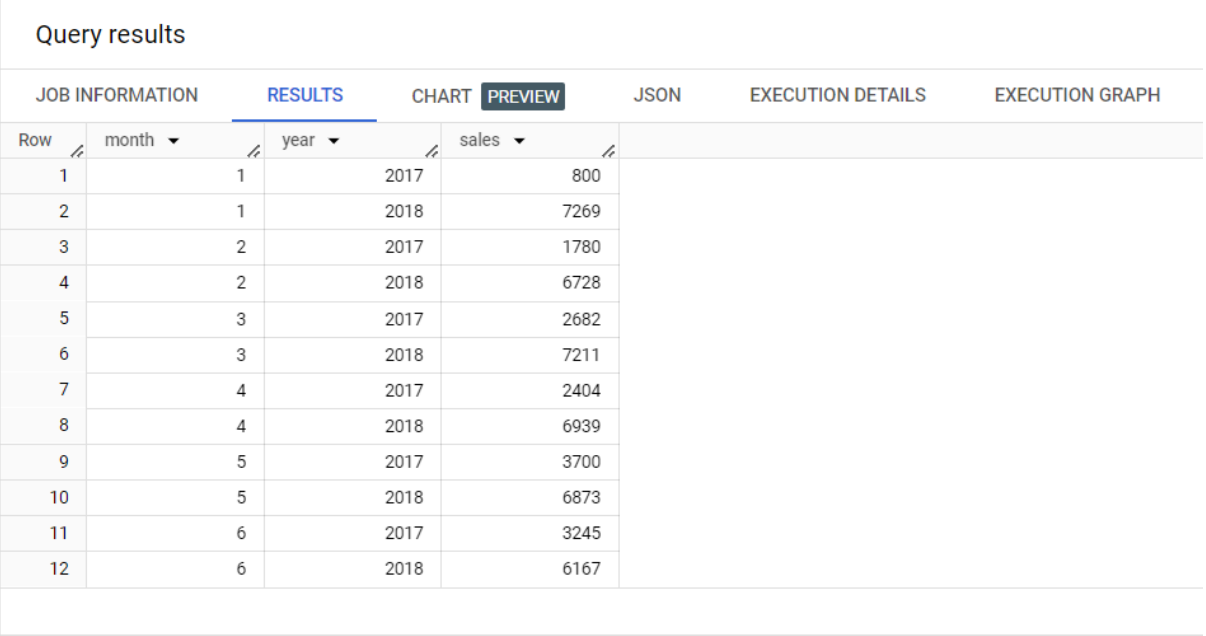
Row	day	total
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Brazilian customers mostly place their order in Afternoon and least during Dawn, which means we can optimize the server resources to best cater the need in afternoon and least during Dawn. Also if there are some offers the best time to launch those are during Dawn hours, if we want to increase sales as organic sales are more during Afternoon.

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

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

```
select extract(month from `order_purchase_timestamp`) as month,
extract(year from `order_purchase_timestamp`) as year,
count(*) as sales
from `windowdate.orders`
group by month, year
order by month, year;
```



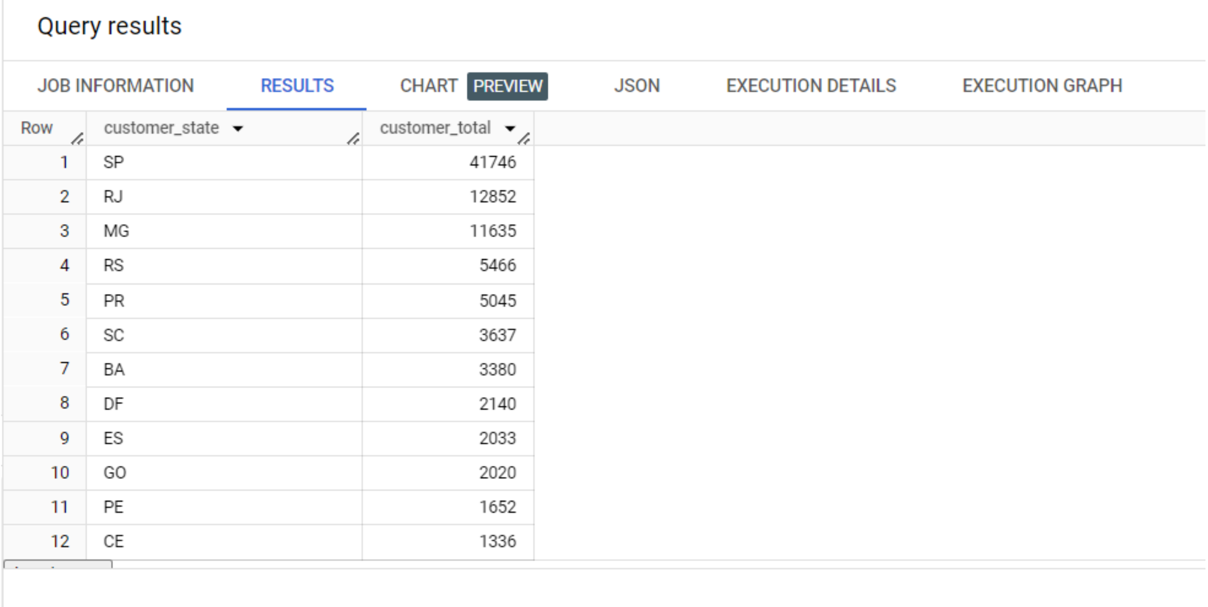
Query results

Row	month	year	sales
1	1	2017	800
2	1	2018	7269
3	2	2017	1780
4	2	2018	6728
5	3	2017	2682
6	3	2018	7211
7	4	2017	2404
8	4	2018	6939
9	5	2017	3700
10	5	2018	6873
11	6	2017	3245
12	6	2018	6167

```
During February and March the the sales almost trippled from 2017 to 2018. During
May and June the the sales almost doubled from 2017 to 2018.
```

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

```
select customer_state, count(customer_id) as customer_total
from `windowdate.customers`
group by customer_state
order by customer_total desc;
```



Query results

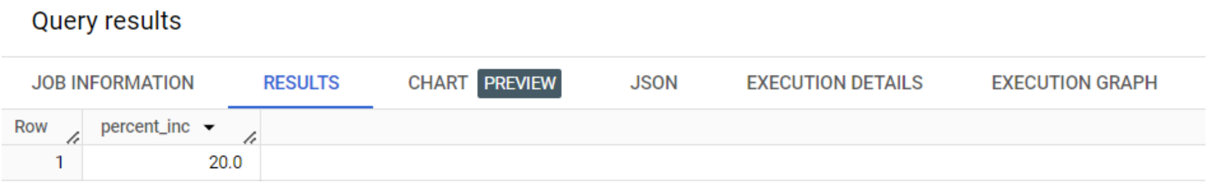
Row	customer_state	customer_total
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336

Major chunk of customer come from SP, followed by RJ and MG. We can find the reason why customers in RJ and MG is drastically reduced as compared to SP, there can be multiple factors both on supply side or demand side, such as presence of competitors or unavailability of products that customers want. These states RJ and MG has potential to increase customer base. Also it may be possible that these states are sparsely populated. So, a better insight would be avaliabe from the percentage of population as the customer with respect to each state.

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

1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
   You can use the "payment_value" column in the payments table to get the cost of orders.

```
with c2017 as
(
  select round(sum (p.payment_value),2) as cost1,
  extract (year from o.order_purchase_timestamp) as years
  from `windowdate.orders` o
  join `windowdate.payments` p
  using (order_id)
  group by years
  having years = 2017
) ,

c2018 as

(
  select round(sum (p.payment_value),2) as cost2,
  extract (year from o.order_purchase_timestamp) as years
  from `windowdate.orders` o
  join `windowdate.payments` p
  using (order_id)
  group by years
  having years = 2018
)

select round(((cost2-cost1)/cost1 *100),2) as percent_inc
from c2018, c2017;
```

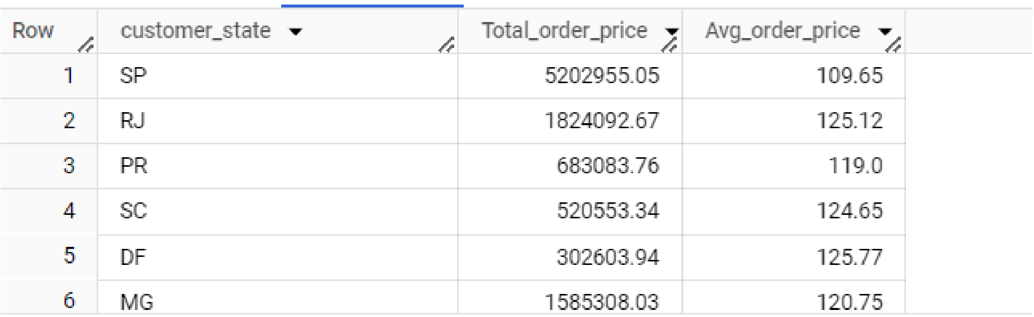


Row	percent_inc
1	20.0

Cost of orders increased to 20 percent. This indicates that customers are spending more on higher end products. As a result the operating cost per order reduces, which means the business should likely see profit due to operations. If the profit is not there then we have to look at the infrastructure cost and human resource cost. As these cost could likely be reduced now.

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

```
select c.customer_state, round(sum(oi.price),2) as Total_order_price
,round(avg(oi.price),2) as Avg_order_price
from `windowdate.order_items` oi
join `windowdate.orders` o  using(order_id)
join `windowdate.customers` c using(customer_id)
group by c.customer_state;
```



Row	customer_state	Total_order_price	Avg_order_price
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	PR	683083.76	119.0
4	SC	520553.34	124.65
5	DF	302603.94	125.77
6	MG	1585308.03	120.75

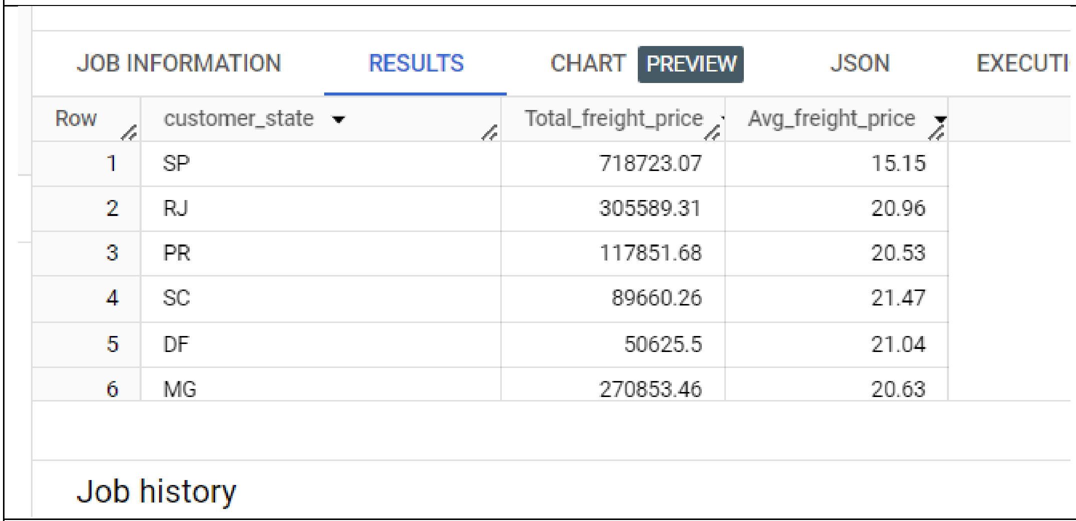
SP has lower average price still total order price is substantially high, means the population base is middle or lower middle class. So, the offers which is lucrative to middle class like 50 percent off, buy2get1, etc would likely increase the sales. A slight increase in price here may also increase the total order price, however the increase should be within the threshold and this threshold value could be found out by experiment on ground.

Similarly for DF, the order price is high ended but total order price is less. So, a slight decrease in prices may increase the sales, or it might be possible that the customers are higher ended, when customer base is higher end they look for quality, so keeping better quality products along with a slight increase in price, likely could increase over all sales.

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

```
select c.customer_state, round(sum(oi.freight_value),2) as Total_freight_price
,round(avg(oi.freight_value),2) as Avg_freight_price
from `windowdate.order_items` oi
join `windowdate.orders` o  using(order_id)
join `windowdate.customers` c using(customer_id)
group by c.customer_state;
```

6.



Row	customer_state	Total_freight_price	Avg_freight_price
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	PR	117851.68	20.53
4	SC	89660.26	21.47
5	DF	50625.5	21.04
6	MG	270853.46	20.63

The average freight price tend to decrease of total freight price increase.

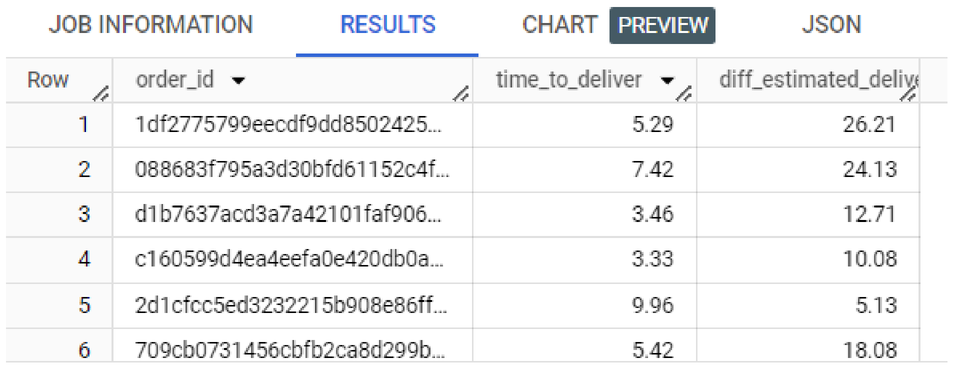
7. **Analysis based on sales, freight and delivery time.**

1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

```
select order_id,round(extract(hour from (order_delivered_carrier_date -
order_purchase_timestamp))/24,2) as time_to_deliver,
round(extract(hour from(order_estimated_delivery_date -
order_delivered_carrier_date))/24,2)  as diff_estimated_delivery
from
windowdate.orders
```

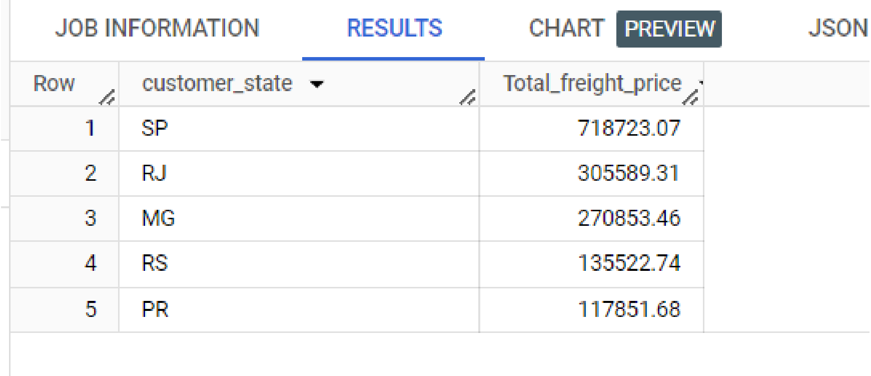


Row	order_id	time_to_deliver	diff_estimated_deliv
1	1df2775799eecdf9dd8502425...	5.29	26.21
2	088683f795a3d30bfd61152c4f...	7.42	24.13
3	d1b7637acd3a7a42101faf906...	3.46	12.71
4	c160599d4ea4eefa0e420db0a...	3.33	10.08
5	2d1cfcc5ed3232215b908e86ff...	9.96	5.13
6	709cb0731456cbfb2ca8d299b...	5.42	18.08

Time to deliver is less than estimated delivery indicates that services are fast and hence satisfied customers.

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

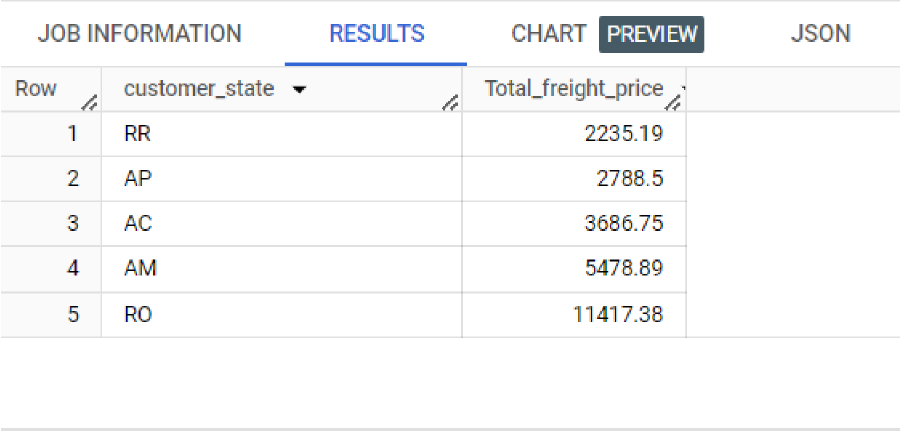
```
select c.customer_state, round(sum(oi.freight_value),2) as Total_freight_price
from `windowdate.order_items` oi
join `windowdate.orders` o  using(order_id)
join `windowdate.customers` c using(customer_id)
group by c.customer_state
order by 2 desc
limit 5
```



Row	customer_state	Total_freight_price
1	SP	718723.07
2	RJ	305589.31
3	MG	270853.46
4	RS	135522.74
5	PR	117851.68

Lowest freight value

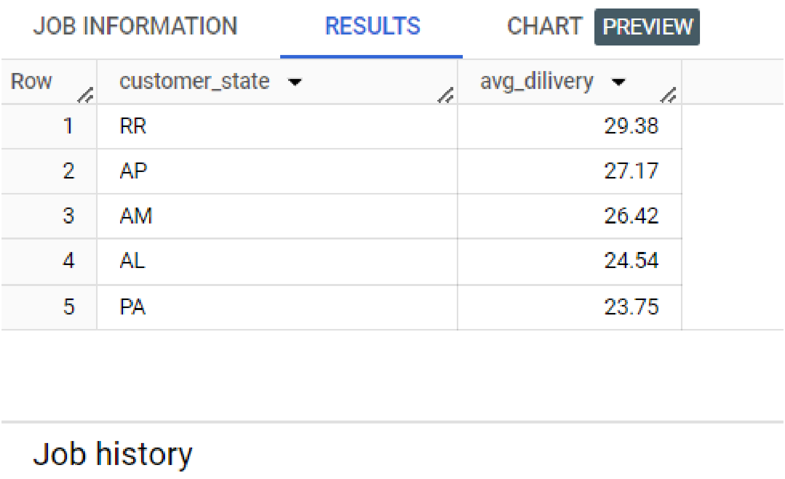
```
select c.customer_state, round(sum(oi.freight_value),2) as Lowest_freight_price
from `windowdate.order_items` oi
join `windowdate.orders` o  using(order_id)
join `windowdate.customers` c using(customer_id)
group by c.customer_state
order by 2 asc
limit 5
```



Row	customer_state	Total_freight_price
1	RR	2235.19
2	AP	2788.5
3	AC	3686.75
4	AM	5478.89
5	RO	11417.38

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

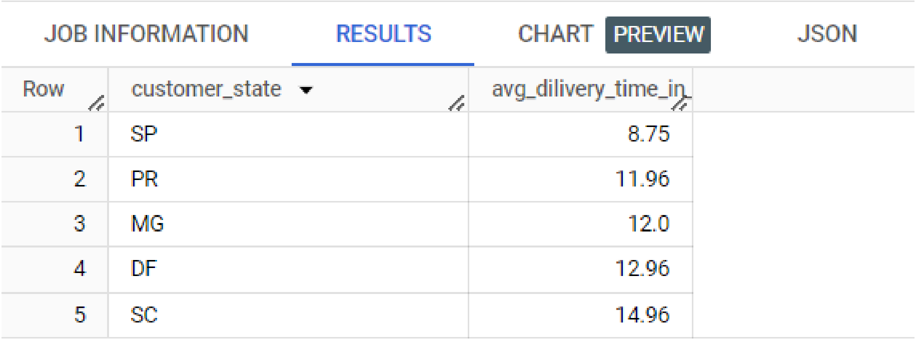
```
select c.customer_state, round(((extract (hour from
avg(order_delivered_customer_date - order_purchase_timestamp)))/24),2) as
avg_dilivery_time_in_days
from
`windowdate.orders` o
join `windowdate.customers` c using (customer_id)
group by 1
order by 2 desc
limit 5

```



Row	customer_state	avg_dilivery
1	RR	29.38
2	AP	27.17
3	AM	26.42
4	AL	24.54
5	PA	23.75

Lowest delivery time

```
select c.customer_state, round(((extract (hour from
avg(order_delivered_customer_date - order_purchase_timestamp)))/24),2) as
avg_dilivery_time_in_days
from
`windowdate.orders` o
join `windowdate.customers` c using (customer_id)
group by 1
order by 2 asc
limit 5
```

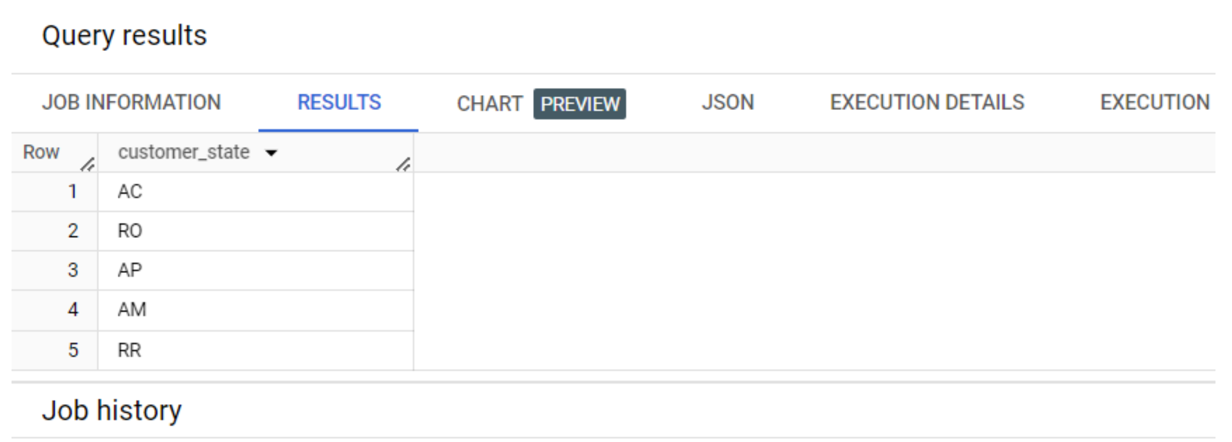


Row	customer_state	avg_dilivery_time_in
1	SP	8.75
2	PR	11.96
3	MG	12.0
4	DF	12.96
5	SC	14.96

1. 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.

```
select customer_state
from
(
  select c.customer_state, round(avg(timestamp_diff(order_estimated_delivery_date,
order_purchase_timestamp, hour)),2) as avg_est,
  round(avg(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp,
hour)),2) as avg_act
  from
  `windowdate.orders` o
  join `windowdate.customers` c using (customer_id)
  group by c.customer_state


) t
group by customer_state
order by max(avg_est- avg_act) desc
limit 5
```



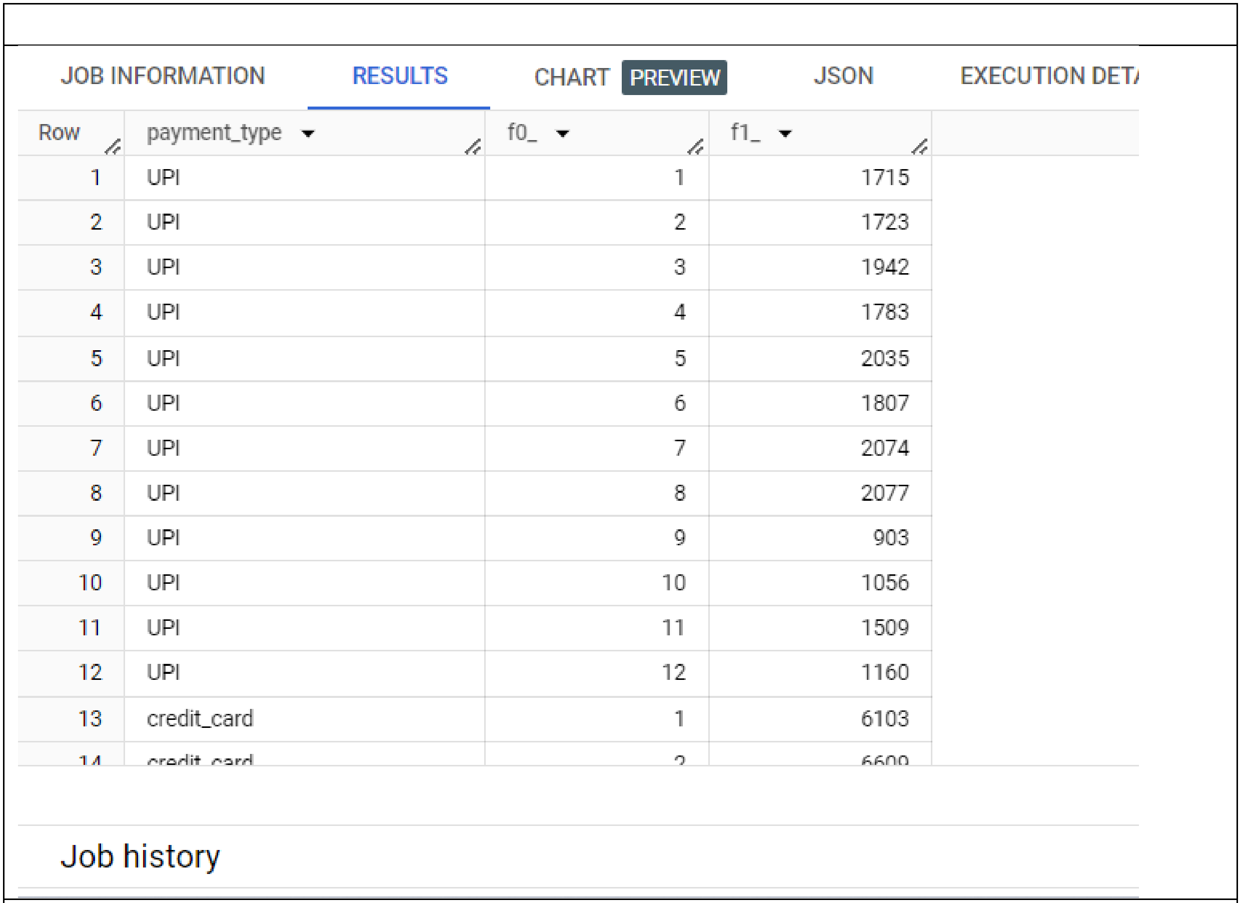
Row	customer_state
1	AC
2	RO
3	AP
4	AM
5	RR

These top 5 states, might be having good customer satisfaction and better logistics control. We can survey these states to find the reason, and can replicated that in other states. Some of the factors include, better warehouse capacity and spatial distribution, optimized logistic control, better human resources, densely populated or population is concentrated in small area, the area of the state can also be small. All these positive factors if any can be replicated in other states, if possible.

21. **Analysis based on the payments:**

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

```
select p.payment_type, extract (month from order_purchase_timestamp),
count(order_id) from
`windowdate.payments` p
join `windowdate.orders` o
using (order_id)
group by 1,2
order by 1,2;
```

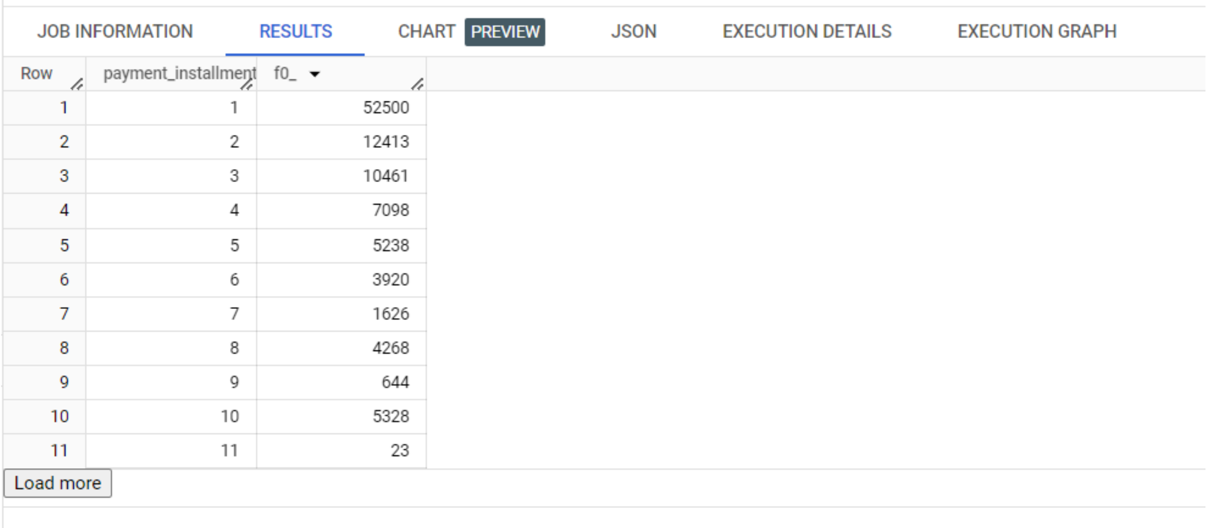


Row	payment_type	f0_	f1_
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6103
14	credit_card	2	6609

UPI is not that popular, Target can tie-up from certain bank(s) popular in that area to increase UPI payments as a result it would be win-win situation of Target as well as bank . As it can increase Financial Inclusion as a result more people would be willing to spend. As Financial inclusion also increases consumerism which leads to in turn increasing consumerism.

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

```
select payment_installments,count(order_id) from
`windowdate.payments`
where order_id not in ( select order_id
                        from `windowdate.payments`
                        where payment_value = 0 or payment_value is null
                      )
                      and payment_installments >=1
group by 1
order by 1
```



Row	payment_installment	f0_
1	1	52500
2	2	12413
3	3	10461
4	4	7098
5	5	5238
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328
11	11	23

Mostly the product is purchased from 1st installment i.e full payment has been made in a go. The next major threshold is a 3rd instalment and then at the 8th and 10th. Target can tie up with any credit card bank, the lucrative credit card offers could be given for 9th and 11th installments

---

Evaluation Criteria (100 points):

1. Initial exploration like checking the structure & characteristics of the data (15 points)
2. In-depth Exploration (15 points)
3. Evolution of E-commerce orders in the Brazil region (10 points)
4. Impact on Economy (20 points)
5. Analysis on sales, freight and delivery time (20 points)
6. Analysis based on the payments (10 points)
7. Actionable Insights & Recommendations (10 points)

Submission Process <IMP>:

Once you're done with the case study...

- Use a Word document to paste your SQL queries along with a screenshot of the first 10 rows from the output.
- List down any valuable insights that you find during the analysis and provide some action items from the company's perspective in order to improve the current situation.
- Convert your solutions doc into a PDF, and upload the same on the platform.

- Please note that after submitting once, you will not be allowed to edit your submission.