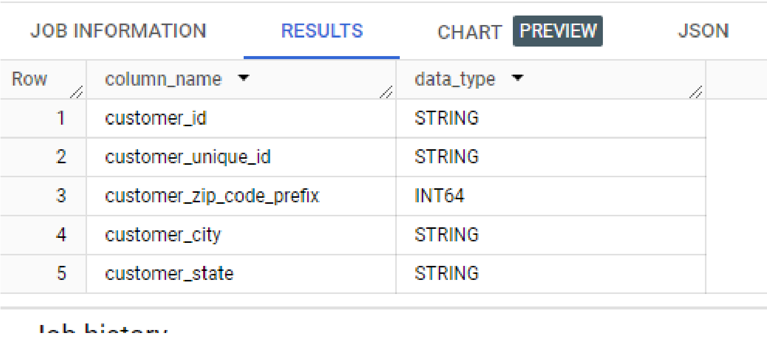
1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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

Ans A:

```
select column_name, data_type
from `teak-catwalk-402416.1casestudy.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers'
```

Screenshot:



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

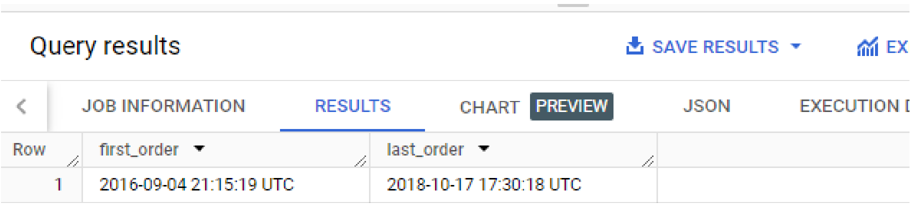
Insight: Utilizing the SQL command we describe customers reveals the data types for all columns in the "customers" table.

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

Ans B:

```
select
min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order
from `first_project_sql.orders`
```

Screenshot :



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

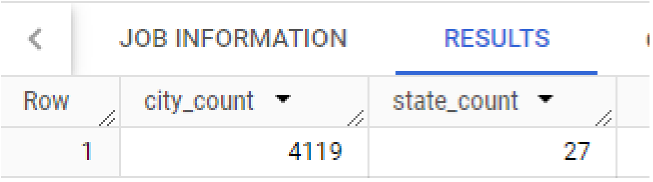
Insights: Utilizing the SQL command we understand that first order were placed in 2016-09–04 and last order were placed in 2018-10-17 by this we got time range between the orders placed

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

Ans C:

```
select
count(distinct customer_city) as city_count,
count(distinct customer_state) as state_count
from `first_project_sql.customers`
```

Screenshot :



Row	city_count	state_count
1	4119	27

Insight: Utilizing the SQL command we understand that 4119 cities and 27 states ordered in a certain period of time.

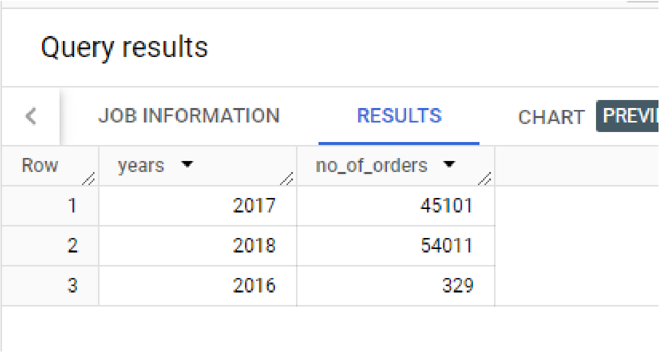
2. In-depth Exploration:

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

Ans A:

```
select
extract(YEAR from order_purchase_timestamp) as years,
count(order_id) as no_of_orders
from `first_project_sql.orders`
group by 1
```

Screenshot :


Query results

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

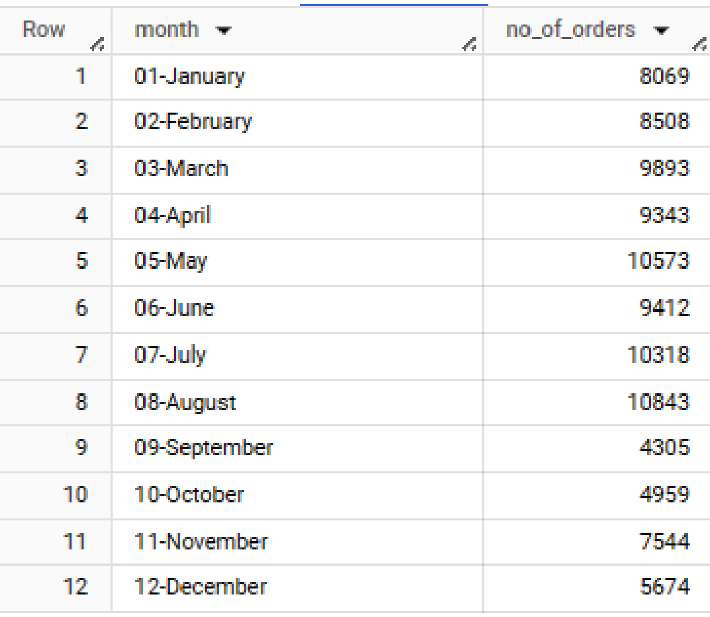
Insights: Utilizing the SQL command we understand that the orders were growing from year to year i,e there were only 329 orders in 2016, 45101 orders in 2017, and 54011 orders in 2018. However data only contains the last 3 months of 2016, so 2016 can be ignored.

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

Ans B:

```
SELECT FORMAT_DATE('%m-%B', DATE(order_purchase_timestamp)) as month,
       COUNT(order_id) as no_of_orders
FROM target.orders
GROUP BY 1
ORDER BY 1
```

Screenshot :



Row	month	no_of_orders
1	01-January	8069
2	02-February	8508
3	03-March	9893
4	04-April	9343
5	05-May	10573
6	06-June	9412
7	07-July	10318
8	08-August	10843
9	09-September	4305
10	10-October	4959
11	11-November	7544
12	12-December	5674

Insights: By using the SQL query, we are able to analyze the orders placed in a given month. The sales are low in last quarter of the year and relatively higher from May to August

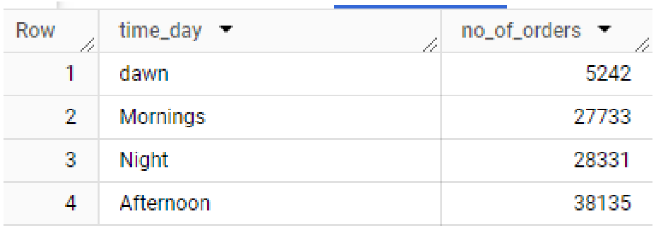
C. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

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

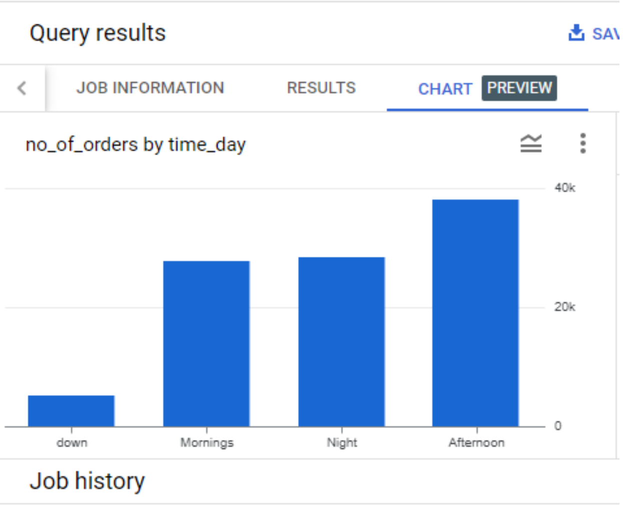
Ans C :

```
select
case when extract (hour from order_purchase_timestamp) between 0 and 6 then 'down'
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 time_day, count(order_id) as no_of_orders
from `first_project_sql.orders`
group by 1
order by 2
```

Screenshot :



Row	time_day	no_of_orders
1	dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135



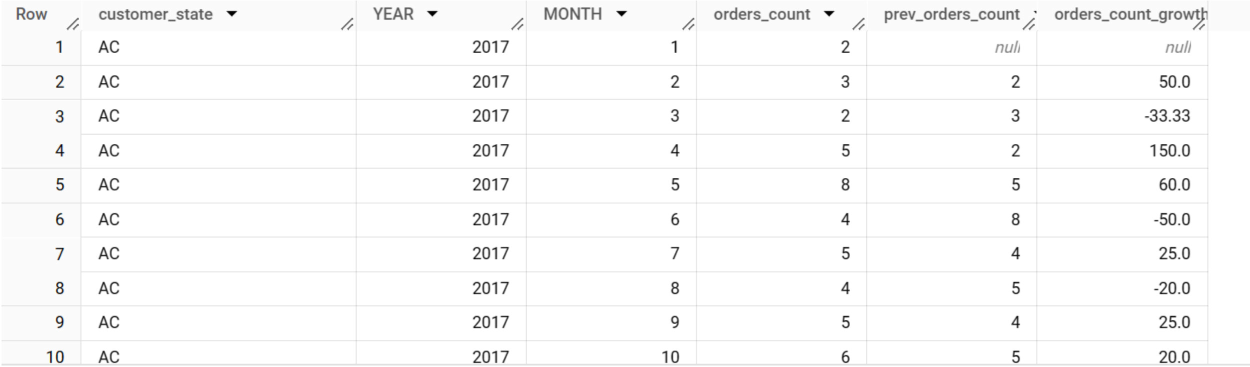
Insights: **We can conclude from the SQL query that there were more orders in the afternoon (i.e., 38135) and less in the dawn (i.e., 5242).**

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

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

```
      SELECT  *,  ROUND(((orders_count  -  prev_orders_count)  /  prev_orders_count)  *  100,  2)
AS orders_count_growth_rate
      FROM
            (SELECT  *,  LAG(orders_count)  OVER(PARTITION  BY  customer_state  ORDER  BY  YEAR,
MONTH)  AS prev_orders_count
            FROM
                (SELECT customer_state,
                        EXTRACT(YEAR FROM order_purchase_timestamp) AS YEAR,
                        EXTRACT(MONTH FROM order_purchase_timestamp) AS MONTH,
                        COUNT(*) AS orders_count
                   FROM `first_project_sql.customers`
                   JOIN `first_project_sql.orders` using (customer_id)
                   Group by 1,2,3))
      order by 1,2,3,6
```

Screenshot :



Row	customer_state	YEAR	MONTH	orders_count	prev_orders_count	orders_count_growth
1	AC	2017	1	2	*null*	*null*
2	AC	2017	2	3	2	50.0
3	AC	2017	3	2	3	-33.33
4	AC	2017	4	5	2	150.0
5	AC	2017	5	8	5	60.0
6	AC	2017	6	4	8	-50.0
7	AC	2017	7	5	4	25.0
8	AC	2017	8	4	5	-20.0
9	AC	2017	9	5	4	25.0
10	AC	2017	10	6	5	20.0

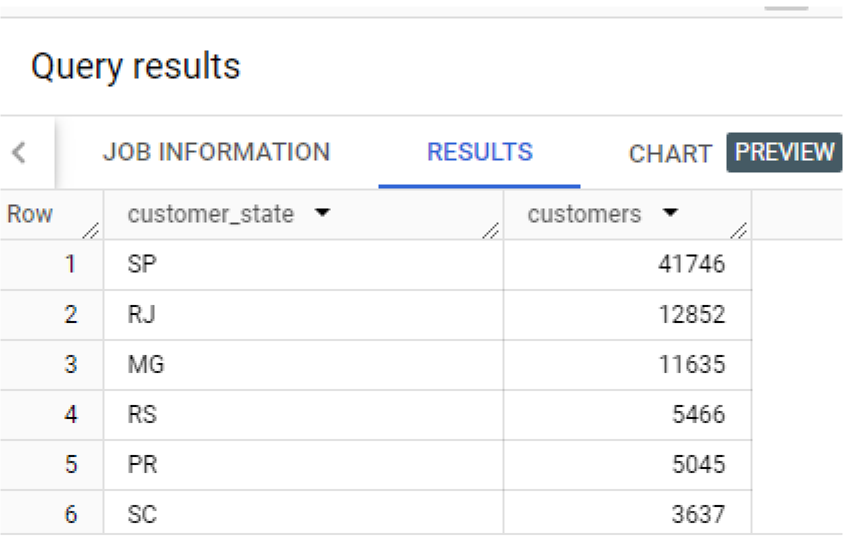
Insights: RR during 2017 January had the highest growth rate of 64 times the previous month. Many states SP, RJ, SC, MG has the lowest growth rate of 99% in 2018

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

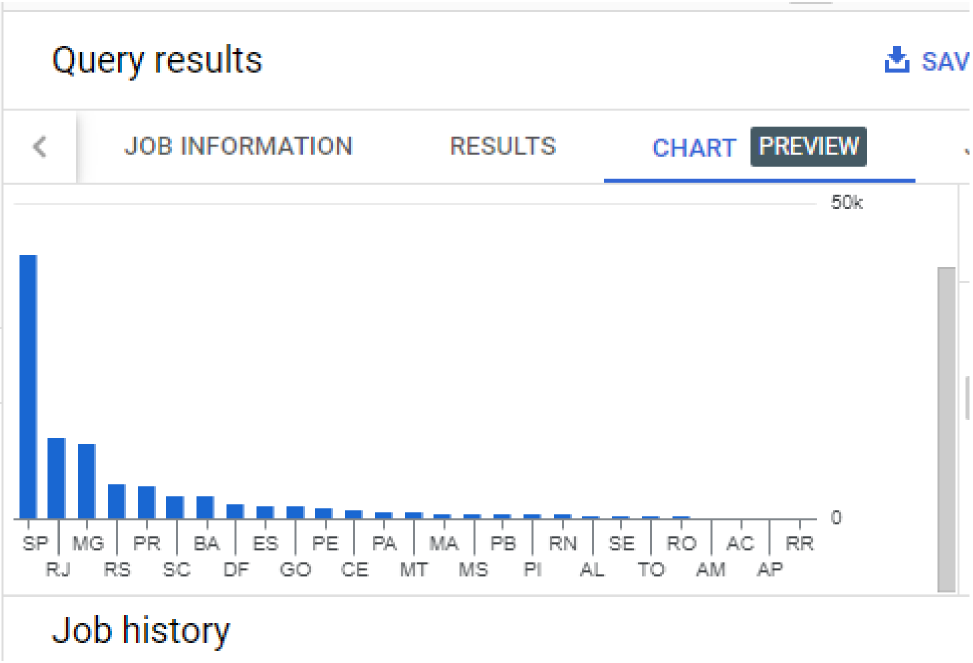
Ans B:

```
select
customer_state, count(customer_id) as customers
from `first_project_sql.customers`
group by 1
order by 2 desc
```

Screenshot :



Row	customer_state	customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637



Insights: By using the SQL command, we were able to determine that SP had the largest number of customers.

4.Impact on Economy: Analyze 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)*.

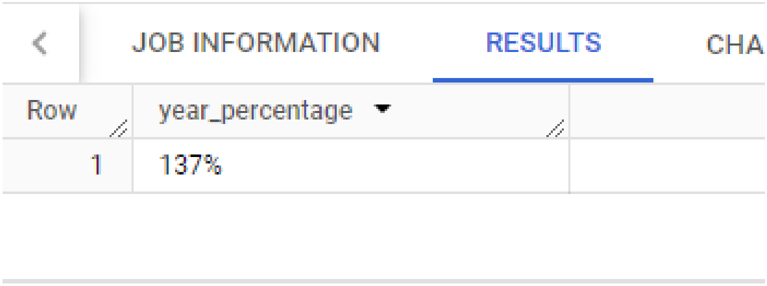
You can use the “payment_value” column in the payments table to get the cost of orders.

Ans A :

```
select distinct concat (round(
((((select sum(payment_value)
from `first_project_sql.payments` join `first_project_sql.orders` using (order_id)
where extract(month from order_purchase_timestamp) between 1 and 8 and
          extract(year from order_purchase_timestamp) = 2018) -
   (select sum(payment_value)
from `first_project_sql.payments` join `first_project_sql.orders` using (order_id)
where extract(month from order_purchase_timestamp) between 1 and 8 and
          extract(year from order_purchase_timestamp) = 2017)) /
(select sum(payment_value)
from `first_project_sql.payments` join `first_project_sql.orders` using (order_id)
where extract(month from order_purchase_timestamp) between 1 and 8 and
                 extract(year from order_purchase_timestamp) = 2017))*100)),'%') as
year_percentage

from `first_project_sql.payments` join `first_project_sql.orders` using(order_id)
```

Screenshot :



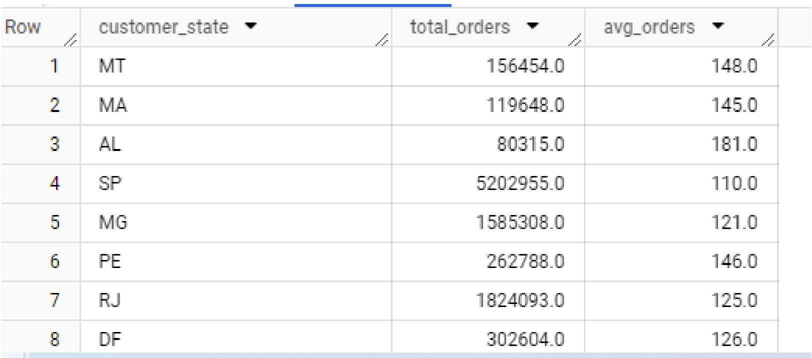
Row	year_percentage
1	137%

Insights: We were able to calculate the 137% increase in order costs from 2017 to 2018 (which only included the months of January through August) by using the SQL command.

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

Ans B:
```
select customer_state,
       round(sum(price)) as total_orders, round(avg(price)) as avg_orders
from `first_project_sql.orders` join `first_project_sql.order_items` using(order_id)
      join `first_project_sql.customers` using(customer_id) group by 1
```

Screenshot :



Row	customer_state	total_orders	avg_orders
1	MT	156454.0	148.0
2	MA	119648.0	145.0
3	AL	80315.0	181.0
4	SP	5202955.0	110.0
5	MG	1585308.0	121.0
6	PE	262788.0	146.0
7	RJ	1824093.0	125.0
8	DF	302604.0	126.0

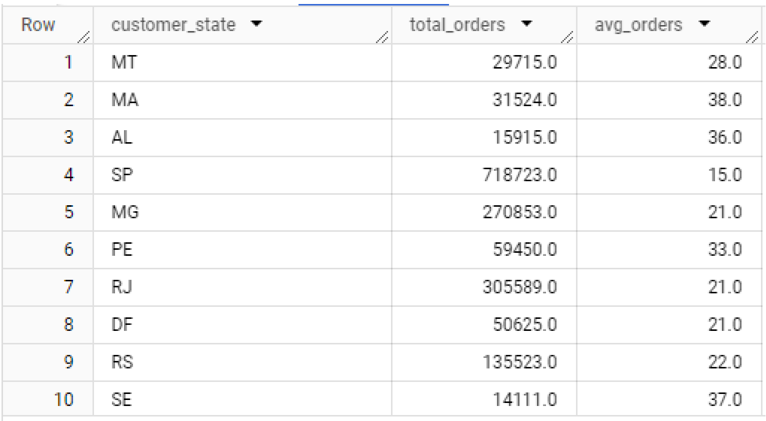
Insights: We were able to determine the average orders per state and the total orders by using the SQL command. Highest sales value was for Sao Paula and lowest was for Roraima. Lowest Average sale was for Sao Paula and highest average sale was for Paraiba state

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

Ans C :

```
select customer_state,
round(sum(freight_value)) as total_orders, round(avg(freight_value)) as avg_orders
from `first_project_sql.orders` join `first_project_sql.order_items`
using(order_id)
join `first_project_sql.customers` using(customer_id)
group by 1
```

Screenshot :



Row	customer_state	total_orders	avg_orders
1	MT	29715.0	28.0
2	MA	31524.0	38.0
3	AL	15915.0	36.0
4	SP	718723.0	15.0
5	MG	270853.0	21.0
6	PE	59450.0	33.0
7	RJ	305589.0	21.0
8	DF	50625.0	21.0
9	RS	135523.0	22.0
10	SE	14111.0	37.0

InsightsInsights:We were able to determine the average orders per state and the total orders by using the SQL command. Highest sales value was for Sao Paula and lowest was for Roraima. Lowest Average sale was for Sao Paula and highest average sale was for Paraiba state

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

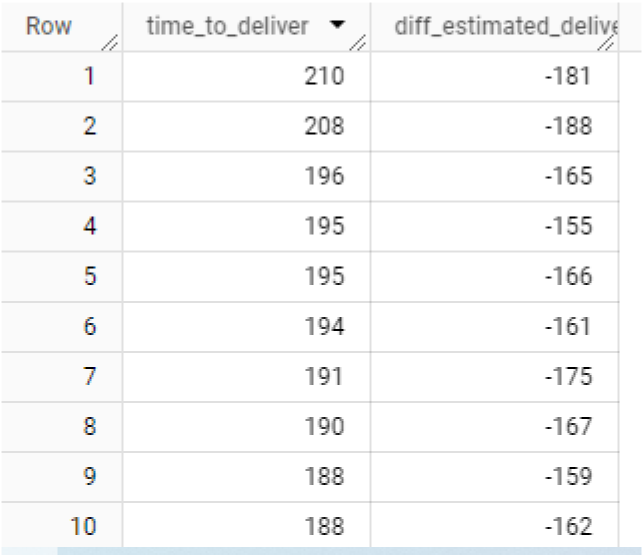
A. Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query. You can calculate the delivery time and the difference between the estimated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

Ans A:

```
select
date_diff  (date(order_delivered_customer_date),  date(order_purchase_timestamp),day)
as time_to_deliver,
date_diff                                        (date(order_estimated_delivery_date),
date(order_delivered_customer_date),day) as diff_estimated_delivery
from `first_project_sql.orders`
order by 1 desc, 2 desc
```

Screenshot :



Row	time_to_deliver	diff_estimated_delive
1	210	-181
2	208	-188
3	196	-165
4	195	-155
5	195	-166
6	194	-161
7	191	-175
8	190	-167
9	188	-159
10	188	-162

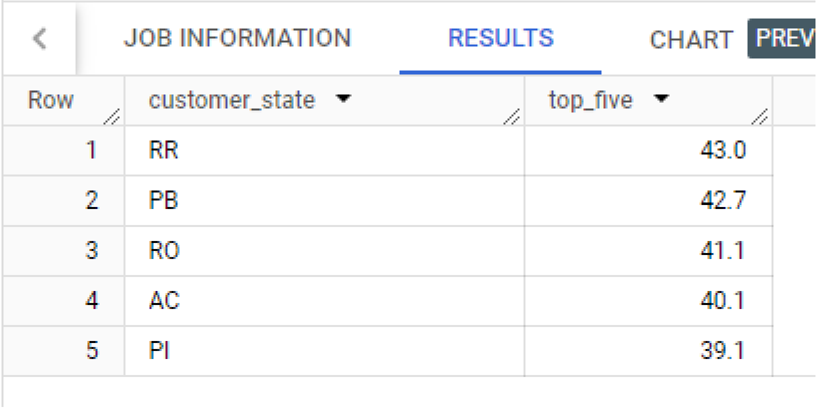
Insights: The delivery time was as high as 210 days differing from the estimated date by a large margin. And quickest delivery was as low as 0 days

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

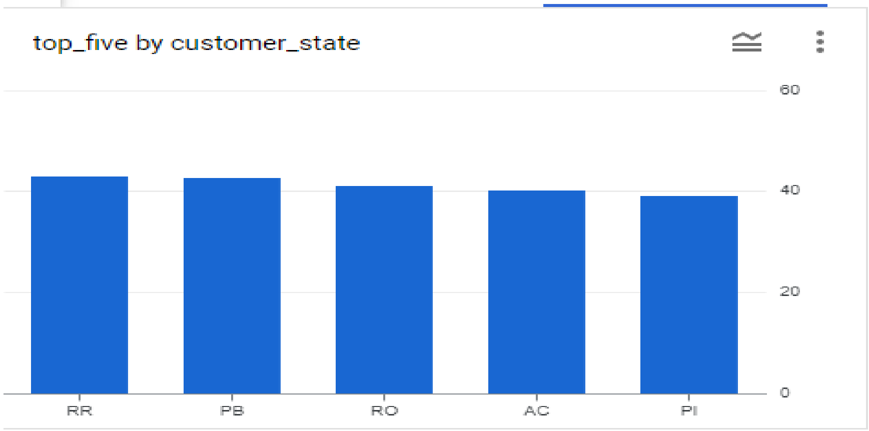
Ans B: 5 states - highest average freight value.

```
select customer_state,
round(avg(freight_value),1) as top_five
from `first_project_sql.orders`
join `first_project_sql.customers` using (customer_id)
join `first_project_sql.order_items` using (order_id)
group by 1
order by 2 desc
limit 5
```

Screenshot :



Row	customer_state	top_five
1	RR	43.0
2	PB	42.7
3	RO	41.1
4	AC	40.1
5	PI	39.1

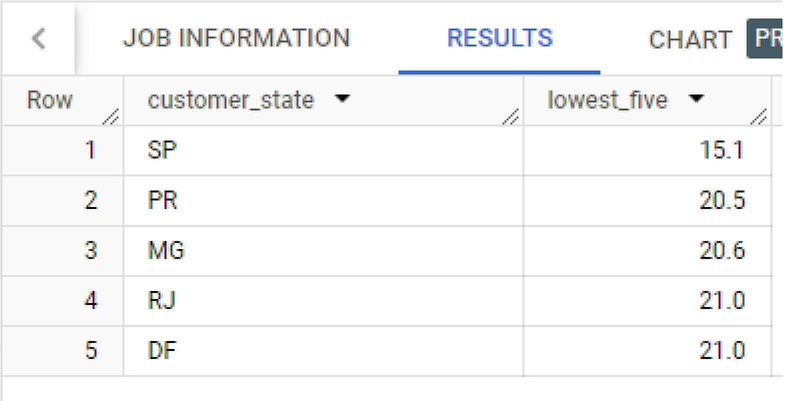


Job history

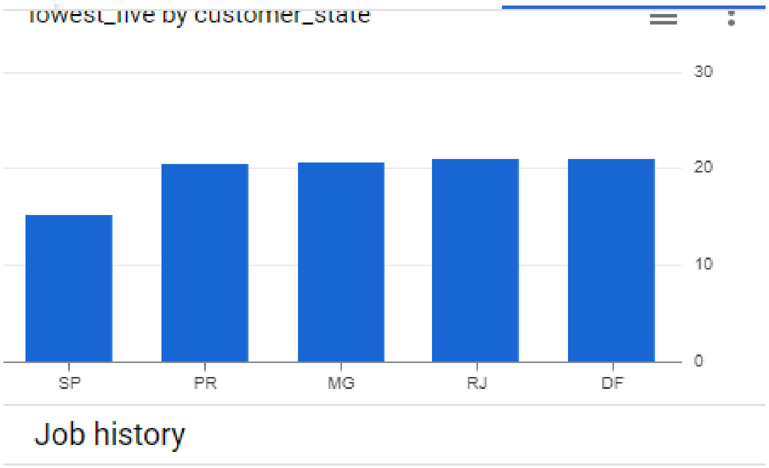
5 states - lowest average freight value.

```
select customer_state,
round(avg(freight_value),1) as lowest_five
from `first_project_sql.orders`
join `first_project_sql.customers` using (customer_id)
join `first_project_sql.order_items` using (order_id)
group by 1
order by 2 asc
limit 5
```

Screenshot :



Row	customer_state	lowest_five
1	SP	15.1
2	PR	20.5
3	MG	20.6
4	RJ	21.0
5	DF	21.0



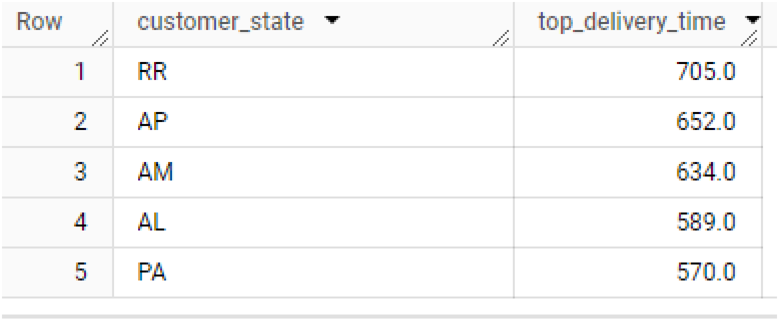
Insights: We noticed that RR has the highest average freight value and SP has the lowest average freight value, respectively, by using the SQL command.

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

Ans c : 5 states - highest average delivery time

```
select customer_state,
round(avg(timestamp_diff(timestamp(order_delivered_customer_date),
timestamp(order_purchase_timestamp),hour))) as top_delivery_time
from `first_project_sql.orders` join `first_project_sql.customers` using
(customer_id)
group by 1
order by 2 desc
limit 5
```

Screenshot :

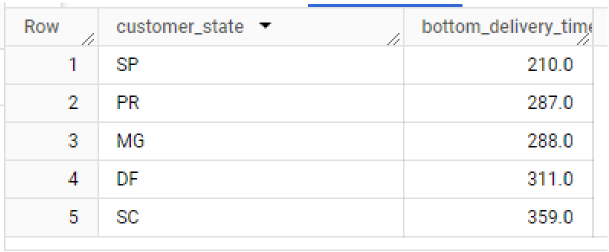


Row	customer_state	top_delivery_time
1	RR	705.0
2	AP	652.0
3	AM	634.0
4	AL	589.0
5	PA	570.0

5 states - lowest average delivery time

```
select customer_state,
round(avg(timestamp_diff(timestamp(order_delivered_customer_date),
timestamp(order_purchase_timestamp),hour))) as bottom_delivery_time
from `first_project_sql.orders` join `first_project_sql.customers` using
(customer_id)
group by 1
order by 2 asc
limit 5
```

Screenshot :



Row	customer_state	bottom_delivery_tim
1	SP	210.0
2	PR	287.0
3	MG	288.0
4	DF	311.0
5	SC	359.0

Insights: Using SQL Command, the top states with the highest and lowest average delivery times are RR and SP, respectively. Results are in hours

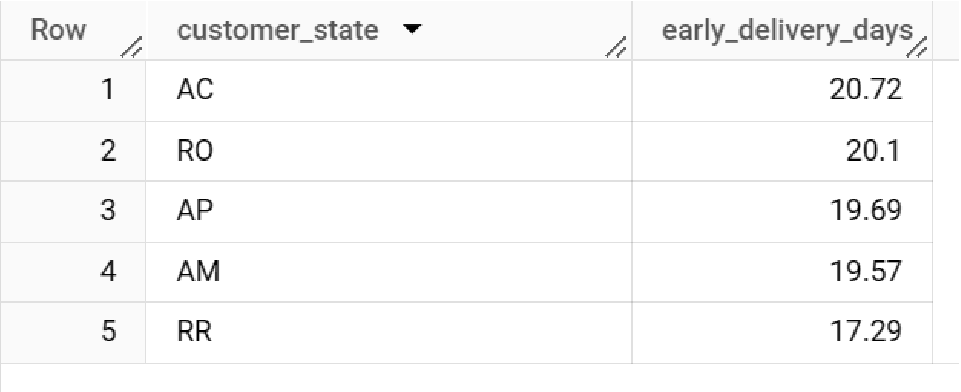
5.D Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Ans 5 D:

```
SELECT
  customer_state,
  round(AVG(date_diff(
DATE(order_estimated_delivery_date),DATE(order_delivered_customer_date),day)),2)     AS
early_delivery_days,
FROM `first_project_sql.orders` join `first_project_sql.customers` using(customer_id)
group by 1
ORDER BY 2 desc
limit 5
```

Screenshot :



Row	customer_state	early_delivery_days
1	AC	20.72
2	RO	20.1
3	AP	19.69
4	AM	19.57
5	RR	17.29

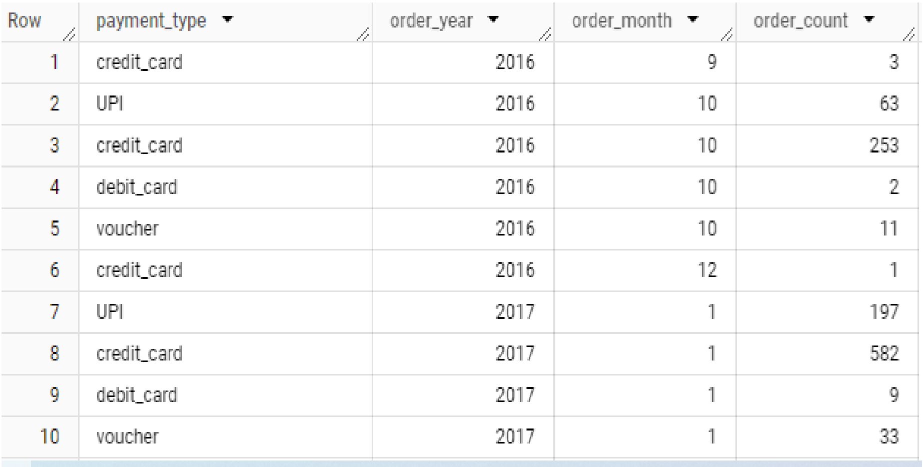
Insights : Order delivery was much faster in above states compared to other states, could be because of faster supply chain or lesser number of orders.

6. Analysis based on the payments:

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

```
Select  payment_type,
extract(year FROM order_purchase_timestamp) as order_year,
extract(Month FROM order_purchase_timestamp) as order_month,
count(distinct order_id) as order_count
from  `first_project_sql.orders`
JOIN `first_project_sql.customers` USING (customer_id)
join `first_project_sql.payments`  using (order_id)
group by 2,3,1
order by 2,3,1
```

Screenshot :



Row	payment_type	order_year	order_month	order_count
1	credit_card	2016	9	3
2	UPI	2016	10	63
3	credit_card	2016	10	253
4	debit_card	2016	10	2
5	voucher	2016	10	11
6	credit_card	2016	12	1
7	UPI	2017	1	197
8	credit_card	2017	1	582
9	debit_card	2017	1	9
10	voucher	2017	1	33

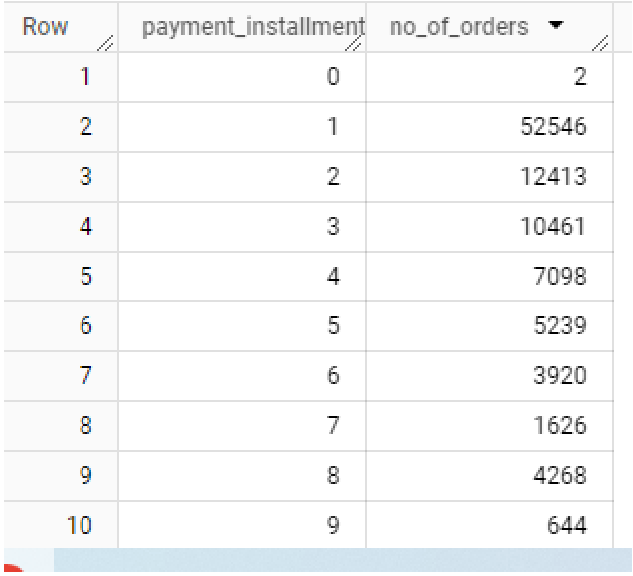
Insights: The highest amount of transactions is done through credit cards.

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

Ans 6 B :

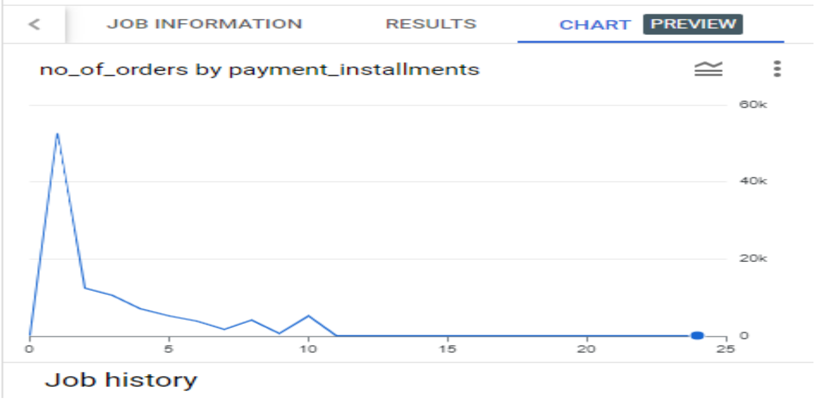
```
select payment_installments, count(order_id) as no_of_orders
from `first_project_sql.payments`
group by 1
order by 1
```

Screenshot :



Row	payment_installment	no_of_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

Insights:



Insights : Most people prefer to have fewer installments. Company can partner with banking partners to lend loans at easy installments.

KEY INSIGHTS(summary)

1. The yearly trend is positive from 2016 to 2018. A dip in sales can be seen from September to December. A rise in sales can be seen from March to August.
2. More sales are made in the afternoon and night, dawn has the least sales.
3. Sao Paulo has the most customers, significantly more than any other state.
4. The order value of an average customer has more than doubled in 2018 from last year
5. States like Roraima, and Paraíba have high freight value.
6. Orders for states like Amapá, and Roraima take a long time to deliver.
7. The highest amount of transactions is done through credit cards.

Recommendation

1. Introducing seasonal/new inventory to lower the impact of the decrease in sales from September.
2. Offering discounts for dawn timings to increase sales.
3. For states like Roraima, and Paraíba with high freight value we can increase the delivery substations to decrease the delivery cost.
4. We can boost the delivery of Amapá, Roraima-like states by changing the means of transportation.
5. As the highest amount of transactions are done through credit cards, we can partner with credit card companies to give special offers to our customers, so that we can increase our sales.
6. A significant amount of people are paying in 1 to 10 installments, which means maybe a lot of people are getting paid on a monthly basis. We can partner with NBFCS and banks to offer them cheap loans, so that they can spend more
7. Target should work on discount pricing strategies before the peak seasons to acquire new customers from the northern regions of Brazil where the customer count is very low, retain the customers in the southern parts of the state, increase sales, and promote new products. This will multiply the profit that Target normally make
8. Target has to develop a good social omnipresence. It should have footprints across all social media platforms to reach new potential customers and sellers. Since the count of customers and sellers is very less in most of the northern regions of Brazil
9. We can see how the orders trajectory is showing a very abrupt increase in orders volume within a very short time. Looking at the overall trend, it is seen that business is picking up very fast in Brazil so companies have to be ready with extra workforce. To avoid high risk, it can consider hiring contractual employees
10. Avg delivery time is quite high for most of those states from where the company is receiving quite less volume of orders, detailed study is needed further for checking the other reasons behind such low volume of orders from majority of states. Huge delivery time can be one of the reasons and we need to work on it. States with highest average delivery time