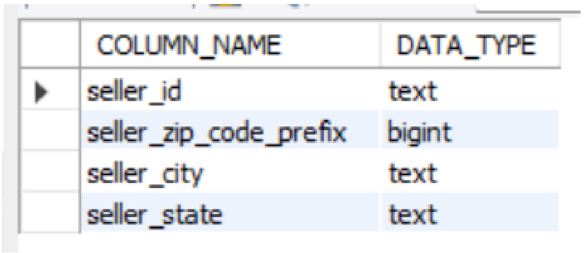
A. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

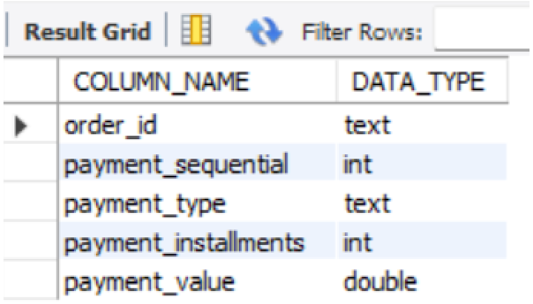
1. Data type of columns in a table

```
select column_name, data_type

from information_schema.columns

where table_schema="sql_project" and table_name="sellers";
```



	COLUMN_NAME	DATA_TYPE
▶	seller_id	text
	seller_zip_code_prefix	bigint
	seller_city	text
	seller_state	text

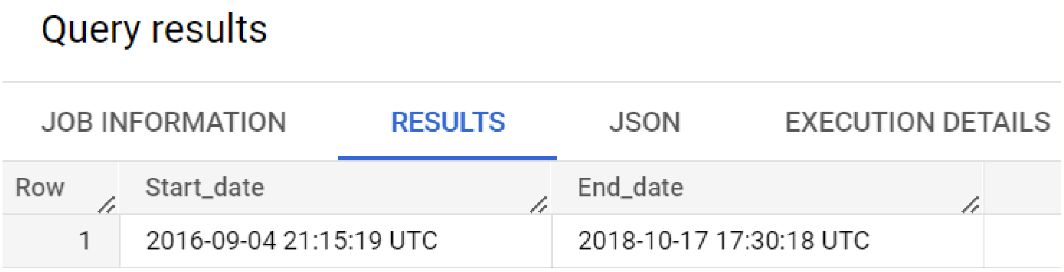
```
select column_name, data_type

from information_schema.columns

where table_schema="sql_project" and table_name="payments";
```



	COLUMN_NAME	DATA_TYPE
▶	order_id	text
	payment_sequential	int
	payment_type	text
	payment_installments	int
	payment_value	double

2. Time period for which the data is given.

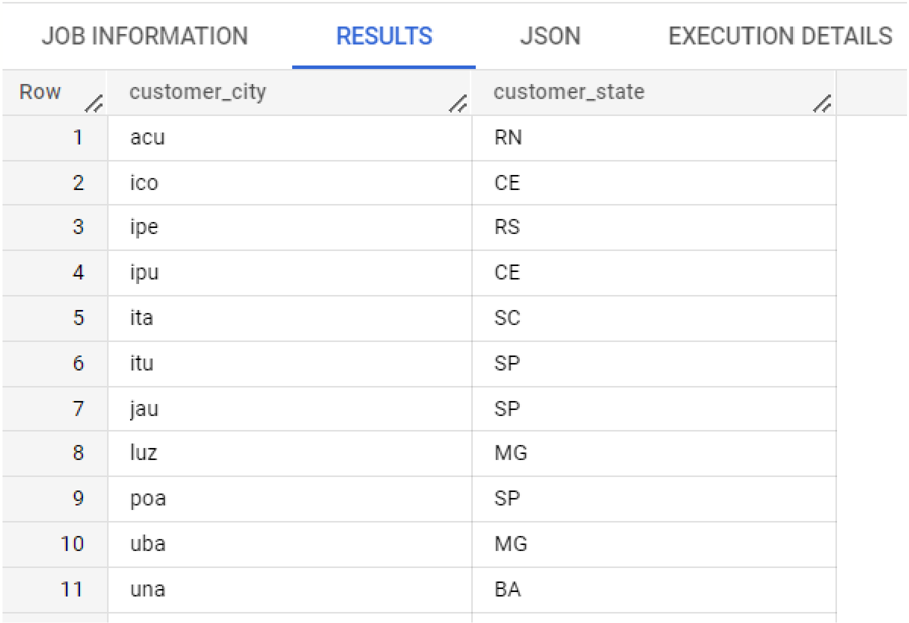
```
Select

Min(order_purchase_timestamp) as Start_date,

Max(order_purchase_timestamp) as End_date

From `Business_Case.orders`
```



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

3. Cities and States of customers ordered during the given period

```
Select DISTINCT

customer_city,

customer_state

From `Business_Case.customers`
```



Row	customer_city	customer_state
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC
6	itu	SP
7	jau	SP
8	luz	MG
9	poa	SP
10	uba	MG
11	una	BA

Insights:

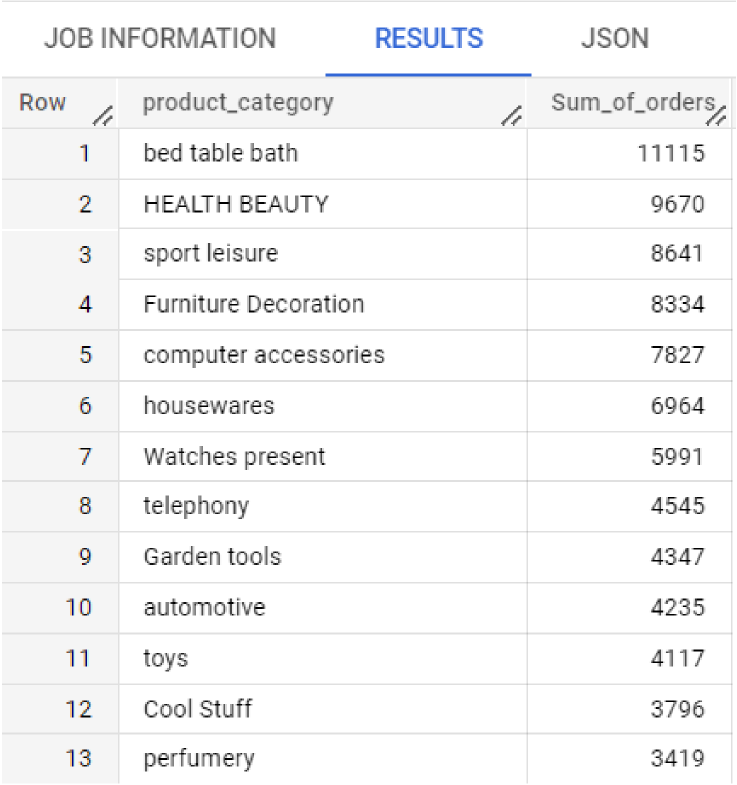
In the dataset, we have int, big int, text(string), double, etc data types. Target has 4119 distinct cities from where people order its products.

B. In-depth Exploration:

1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Most ordered product:

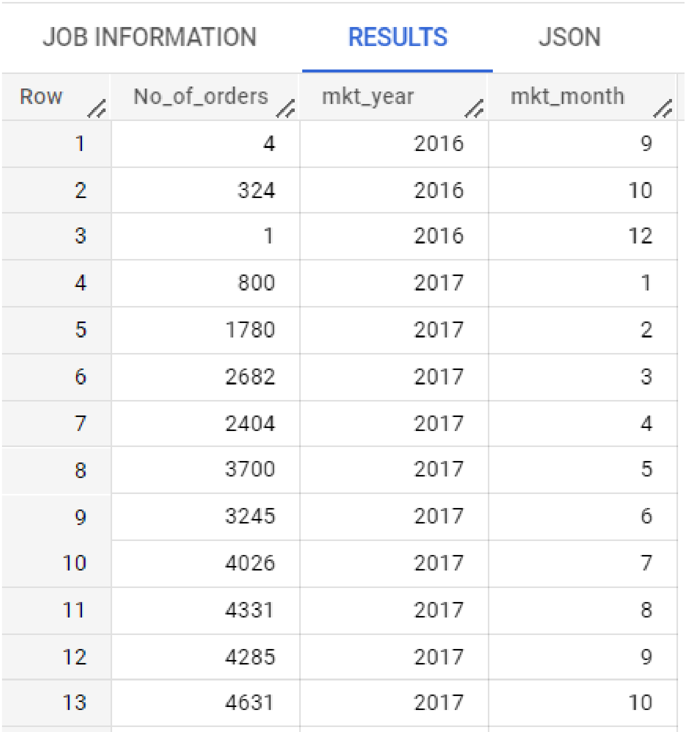
```
Select

p.product_category,

Count(p.product_category) as Sum_of_orders,

From `Business_Case.orders` as o

Join `Business_Case.order_items` as oi

ON o.order_id=oi.order_id

Join `Business_Case.products` as p

ON oi.product_id=p.product_id

Group By p.product_category

Order by Sum_of_orders DESC
```



Row	product_category	Sum_of_orders
1	bed table bath	11115
2	HEALTH BEAUTY	9670
3	sport leisure	8641
4	Furniture Decoration	8334
5	computer accessories	7827
6	housewares	6964
7	Watches present	5991
8	telephony	4545
9	Garden tools	4347
10	automotive	4235
11	toys	4117
12	Cool Stuff	3796
13	perfumery	3419

No. of orders each month each year:

```
SELECT

COUNT(DISTINCT(order_id)) as No_of_orders,

EXTRACT(YEAR FROM order_purchase_timestamp) AS mkt_year,

EXTRACT(month FROM order_purchase_timestamp) AS mkt_month

FROM `Business_Case.orders`

GROUP BY mkt_year,mkt_month

ORDER BY mkt_year,mkt_month
```

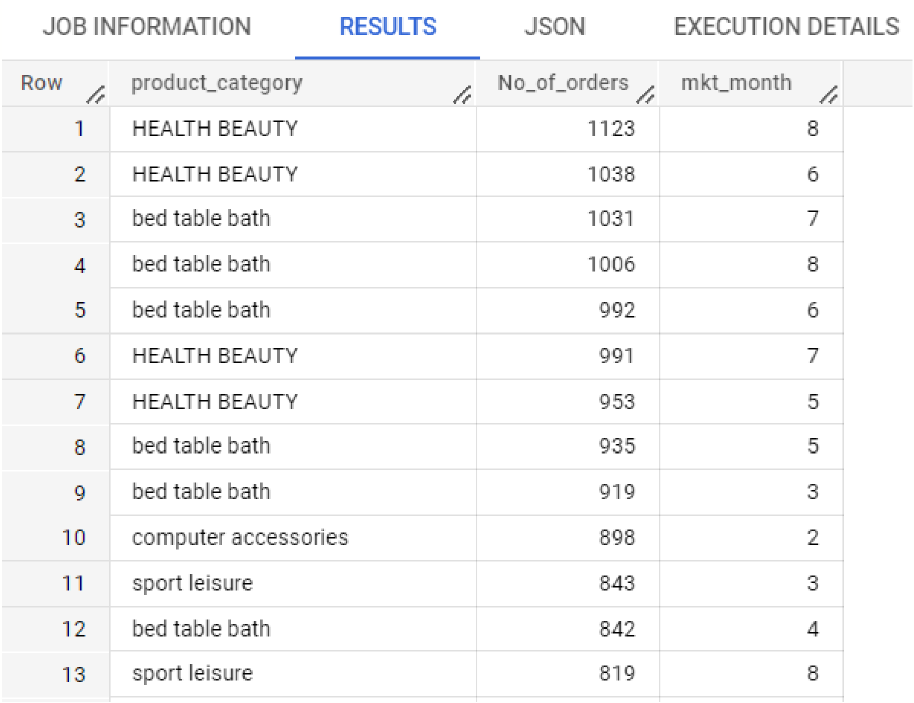


Row	No_of_orders	mkt_year	mkt_month
1	4	2016	9
2	324	2016	10
3	1	2016	12
4	800	2017	1
5	1780	2017	2
6	2682	2017	3
7	2404	2017	4
8	3700	2017	5
9	3245	2017	6
10	4026	2017	7
11	4331	2017	8
12	4285	2017	9
13	4631	2017	10

No. of orders per month with the product category:

```
SELECT

p.product_category,

COUNT(DISTINCT(o.order_id)) as No_of_orders,

EXTRACT(month FROM o.order_purchase_timestamp) AS mkt_month
```

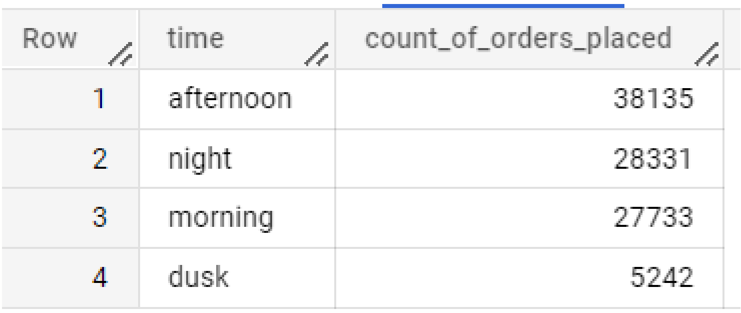
```
From `Business_Case.orders` as o

Join `Business_Case.order_items` as oi

ON o.order_id=oi.order_id

Join `Business_Case.products` as p

ON oi.product_id=p.product_id

GROUP BY mkt_month,p.product_category

ORDER BY No_of_orders desc
```



Row	product_category	No_of_orders	mkt_month
1	HEALTH BEAUTY	1123	8
2	HEALTH BEAUTY	1038	6
3	bed table bath	1031	7
4	bed table bath	1006	8
5	bed table bath	992	6
6	HEALTH BEAUTY	991	7
7	HEALTH BEAUTY	953	5
8	bed table bath	935	5
9	bed table bath	919	3
10	computer accessories	898	2
11	sport leisure	843	3
12	bed table bath	842	4
13	sport leisure	819	8

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
SELECT

CASE

  WHEN hours in (0,1,2,3,4,5,6) THEN "dusk"

  WHEN hours in (7,8,9,10,11,12) THEN "morning"
  WHEN hours in (13,14,15,16,17,18) THEN "afternoon"

  WHEN hours in (19,20,21,22,23) THEN "night"

END as time,

COUNT(t.hours) AS count_of_orders_placed

FROM

(SELECT *, EXTRACT(hour FROM order_purchase_timestamp) AS hours,

FROM `Business_Case.orders`) as t

GROUP BY time

ORDER BY count_of_orders_placed DESC
```



Row	time	count_of_orders_placed
1	afternoon	38135
2	night	28331
3	morning	27733
4	dusk	5242

Insights:

- The top 2 most bought product category is 'bed table bath' and 'health beauty' product.
- Number of orders has been increasing year by year and month by month.
- Customers' prefer to buy products in the afternoon as we can see most of the orders are placed at that time.

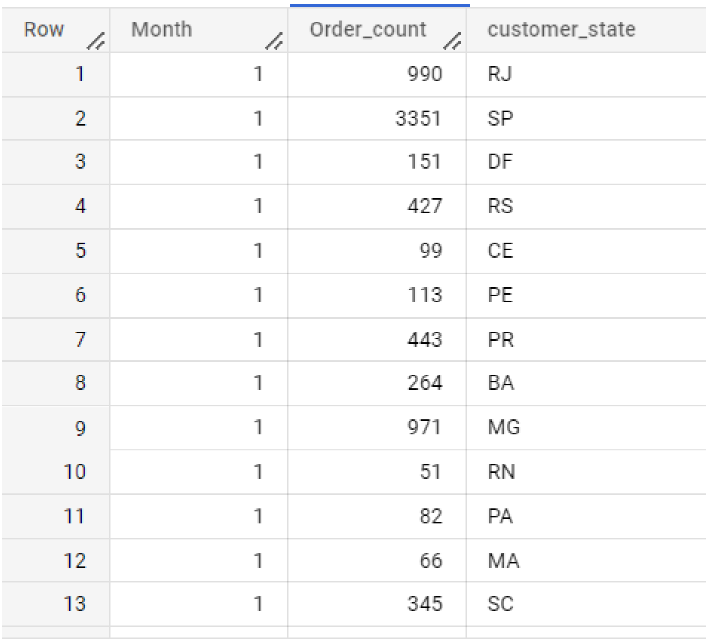
Recommendation:

- We may run sales campaigns to increase sales during the other hours like morning and night on the top 5 selling product categories.
- We may also flash special discounts to our customers during the afternoon on the product categories that have low sales.

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

1. Get month-on-month orders by states

```
SELECT

EXTRACT(month FROM o.order_purchase_timestamp) as Month,

COUNT(o.order_id) as Order_count,

c.customer_state

FROM `Business_Case.orders`as o

JOIN `Business_Case.customers` as c

ON o.customer_id=c.customer_id

GROUP BY month, c.customer_state

ORDER BY month
```

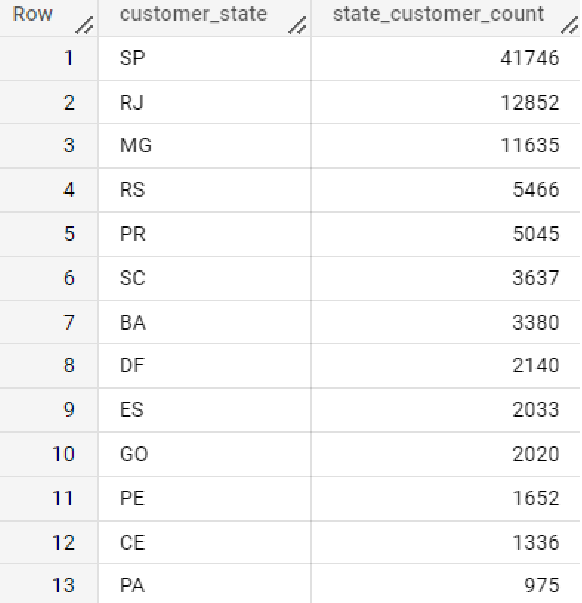


Row	Month	Order_count	customer_state
1	1	990	RJ
2	1	3351	SP
3	1	151	DF
4	1	427	RS
5	1	99	CE
6	1	113	PE
7	1	443	PR
8	1	264	BA
9	1	971	MG
10	1	51	RN
11	1	82	PA
12	1	66	MA
13	1	345	SC

2. Distribution of customers across the states in Brazil

```
SELECT

customer_state,

COUNT(customer_id) as state_customer_count

FROM `Business_Case.customers`
```

```
GROUP BY customer_state

ORDER BY state_customer_count DESC
```



Row	customer_state	state_customer_count
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
13	PA	975

Insights:

- We have most of our customers from ‘SP’ state.

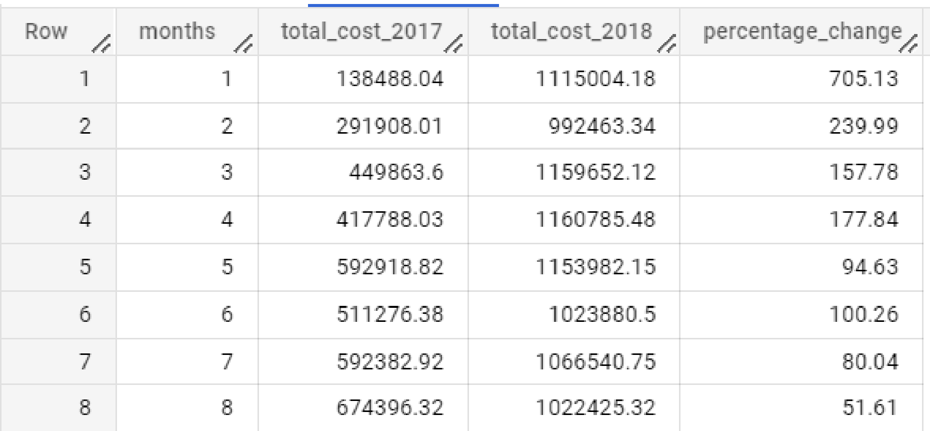
Recommendation:

- To increase customers and sales in other states, we can do a market research as to what kind of products people b in other states and target the customer accordingly. We can study the age group, lifestyle, and level of income as well to make a concrete plan.

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

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

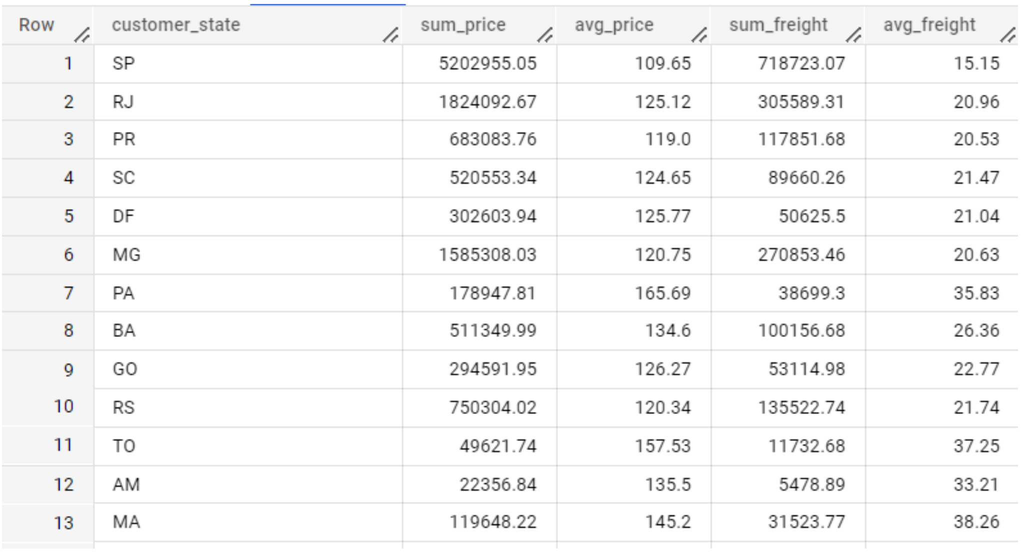
```
WITH

data_2017 as

(SELECT

EXTRACT(Month FROM t1.date) as month,
ROUND(SUM(t1.payment_value),2) as total_cost
FROM (SELECT
EXTRACT(DATE FROM o.order_purchase_timestamp) as date,
p.payment_value
FROM `Business_Case.orders` as o
JOIN `Business_Case.payments` as p
ON o.order_id=p.order_id) as t1
WHERE t1.date BETWEEN "2017-01-01" AND "2017-08-31"
GROUP BY month),
data_2018 as
(SELECT
EXTRACT(Month FROM t1.date) as month,
ROUND(SUM(t1.payment_value),2) as total_cost
FROM (SELECT
EXTRACT(DATE FROM o.order_purchase_timestamp) as date,
p.payment_value
FROM `Business_Case.orders` as o
JOIN `Business_Case.payments` as p
ON o.order_id=p.order_id) as t1
WHERE t1.date BETWEEN "2018-01-01" AND "2018-08-31"
GROUP BY month)
SELECT
d1.month as months,
d1.total_cost as total_cost_2017,
d2.total_cost as total_cost_2018,
ROUND((d2.total_cost-d1.total_cost)*100/d1.total_cost,2) as percentage_change

FROM data_2017 as d1 join data_2018 as d2

ON d1.month=d2.month

order by d1.month
```



Row	months	total_cost_2017	total_cost_2018	percentage_change
1	1	138488.04	1115004.18	705.13
2	2	291908.01	992463.34	239.99
3	3	449863.6	1159652.12	157.78
4	4	417788.03	1160785.48	177.84
5	5	592918.82	1153982.15	94.63
6	6	511276.38	1023880.5	100.26
7	7	592382.92	1066540.75	80.04
8	8	674396.32	1022425.32	51.61

2. Mean & Sum of price and freight value by customer state

```
 SELECT

c.customer_state,

ROUND(SUM(ot.price),2) AS sum_price,

ROUND(AVG(ot.price),2) AS avg_price,

ROUND(SUM(ot.freight_value),2) AS sum_freight,

ROUND(AVG(ot.freight_value),2) AS avg_freight

FROM `Business_Case.order_items` AS ot

LEFT JOIN `Business_Case.orders` AS o

ON ot.order_id=o.order_id

LEFT JOIN `Business_Case.customers` AS c

ON o.customer_id=c.customer_id

GROUP BY c.customer_state
```



Row	customer_state	sum_price	avg_price	sum_freight	avg_freight
1	SP	5202955.05	109.65	718723.07	15.15
2	RJ	1824092.67	125.12	305589.31	20.96
3	PR	683083.76	119.0	117851.68	20.53
4	SC	520553.34	124.65	89660.26	21.47
5	DF	302603.94	125.77	50625.5	21.04
6	MG	1585308.03	120.75	270853.46	20.63
7	PA	178947.81	165.69	38699.3	35.83
8	BA	511349.99	134.6	100156.68	26.36
9	GO	294591.95	126.27	53114.98	22.77
10	RS	750304.02	120.34	135522.74	21.74
11	TO	49621.74	157.53	11732.68	37.25
12	AM	22356.84	135.5	5478.89	33.21
13	MA	119648.22	145.2	31523.77	38.26

Insights:

- We have seen a percentage increase in the cost of orders from 2017 to 2018. However, the percentage is gradually decreasing month on month.

Recommendation:

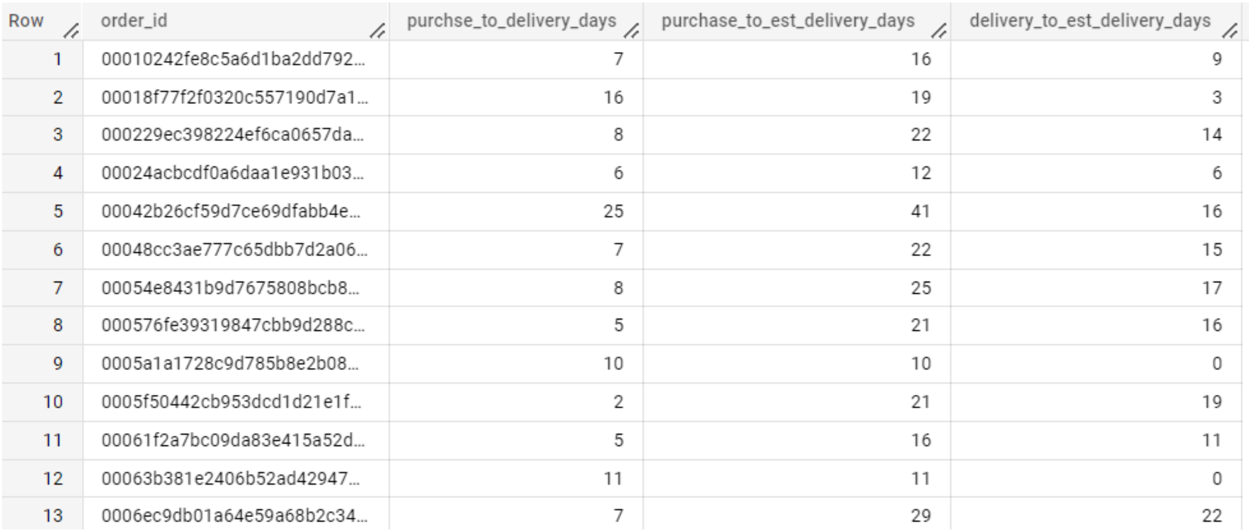
- In the freight and price comparison, we can see that price increases as the freight of the product increases for every state. We can improve our distribution network and reduce our transport costs which will help bring down the cost of the order. This will facilitate quick delivery with lower cost and encourage customers.

E. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
SELECT

order_id,

DATE_DIFF(t.delivery_date,t.purchase_date, DAY) AS purchse_to_delivery_days,

DATE_DIFF(t.est_delivery_date,t.purchase_date, DAY) AS purchase_to_est_delivery_days,

DATE_DIFF(t.est_delivery_date,t.delivery_date, DAY) AS delivery_to_est_delivery_days
```

```
FROM

(SELECT

order_id,

EXTRACT(date FROM order_purchase_timestamp) as purchase_date,

EXTRACT(date FROM order_delivered_customer_date) as delivery_date,

EXTRACT(date FROM order_estimated_delivery_date) as est_delivery_date

FROM `Business_Case.orders`

)AS t

ORDER BY order_id
```

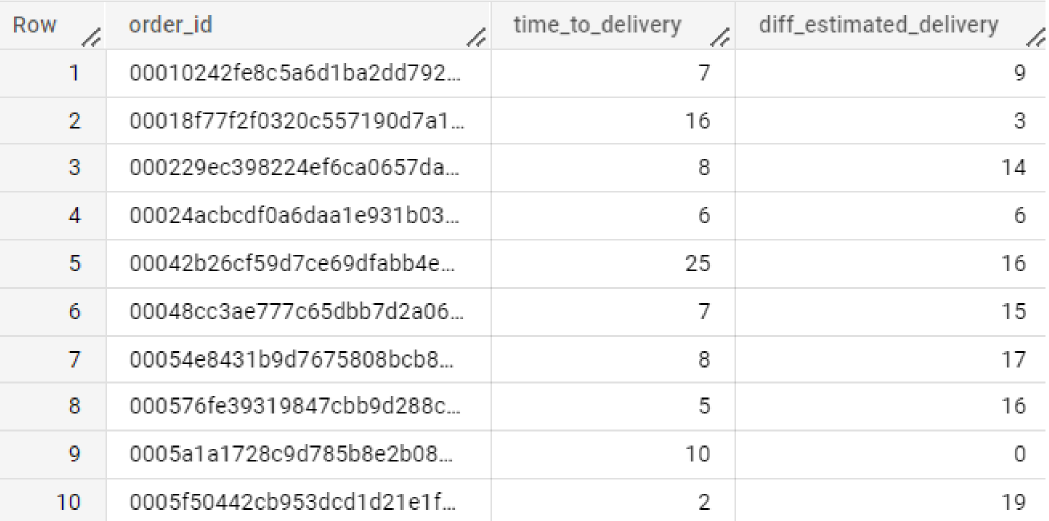


Row	order_id	purchse_to_delivery_days	purchase_to_est_delivery_days	delivery_to_est_delivery_days
1	00010242fe8c5a6d1ba2dd792...	7	16	9
2	00018f77f2f0320c557190d7a1...	16	19	3
3	000229ec398224ef6ca0657da...	8	22	14
4	00024acbcdf0a6daa1e931b03...	6	12	6
5	00042b26cf59d7ce69dfabb4e...	25	41	16
6	00048cc3ae777c65dbb7d2a06...	7	22	15
7	00054e8431b9d7675808bcb8...	8	25	17
8	000576fe39319847cbb9d288c...	5	21	16
9	0005a1a1728c9d785b8e2b08...	10	10	0
10	0005f50442cb953dcd1d21e1f...	2	21	19
11	00061f2a7bc09da83e415a52d...	5	16	11
12	00063b381e2406b52ad42947...	11	11	0
13	0006ec9db01a64e59a68b2c34...	7	29	22

2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
    a. time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
    b. diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
SELECT

order_id,

DATE_DIFF(t.delivery_date,t.purchase_date, DAY) AS time_to_delivery,

DATE_DIFF(t.est_delivery_date,t.delivery_date, DAY) AS diff_estimated_delivery
```

```
FROM

(SELECT

order_id,

EXTRACT(date FROM order_purchase_timestamp) as purchase_date,

EXTRACT(date FROM order_delivered_customer_date) as delivery_date,

EXTRACT(date FROM order_estimated_delivery_date) as est_delivery_date

FROM `Business_Case.orders`

)AS t

ORDER BY order_id
```

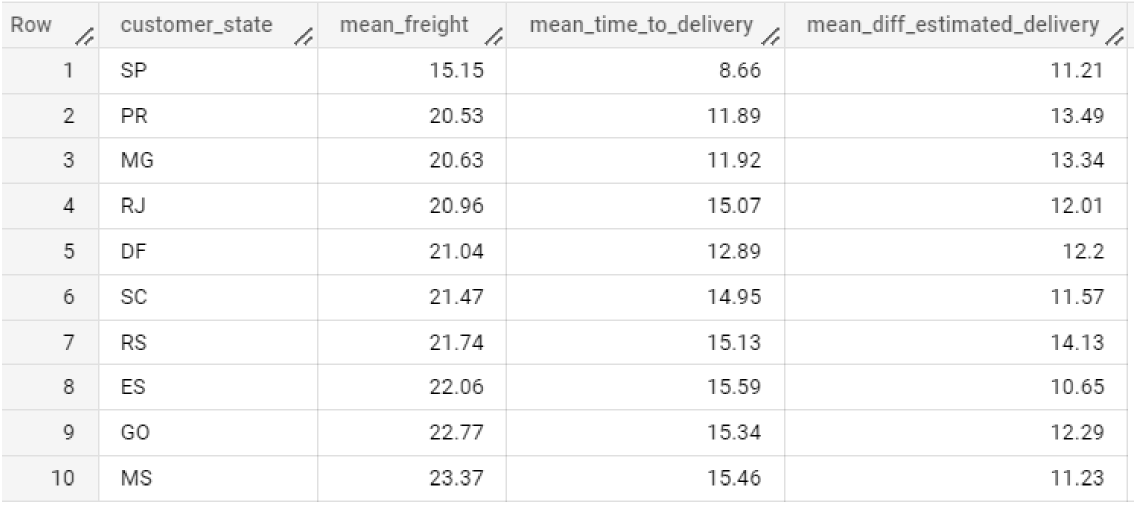


Row	order_id	time_to_delivery	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	9
2	00018f77f2f0320c557190d7a1...	16	3
3	000229ec398224ef6ca0657da...	8	14
4	00024acbcdf0a6daa1e931b03...	6	6
5	00042b26cf59d7ce69dfabb4e...	25	16
6	00048cc3ae777c65dbb7d2a06...	7	15
7	00054e8431b9d7675808bcb8...	8	17
8	000576fe39319847cbb9d288c...	5	16
9	0005a1a1728c9d785b8e2b08...	10	0
10	0005f50442cb953dcd1d21e1f...	2	19

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
SELECT

c.customer_state,

ROUND(AVG(ot.freight_value),2) AS mean_freight,

ROUND(AVG(DATE_DIFF(t.delivery_date,t.purchase_date, DAY)),2) AS mean_time_to_delivery,

ROUND(AVG(DATE_DIFF(t.est_delivery_date,t.delivery_date, DAY)),2) AS mean_diff_estimated_delivery
```

```
FROM

(SELECT

order_id,

customer_id,

EXTRACT(date FROM order_purchase_timestamp) as purchase_date,

EXTRACT(date FROM order_delivered_customer_date) as delivery_date,

EXTRACT(date FROM order_estimated_delivery_date) as est_delivery_date

FROM `Business_Case.orders`)AS t

JOIN `Business_Case.order_items` as ot

ON t.order_id=ot.order_id

JOIN `Business_Case.customers` as c

ON t.customer_id=c.customer_id

GROUP BY c.customer_state

ORDER BY mean_freight

LIMIT 10
```



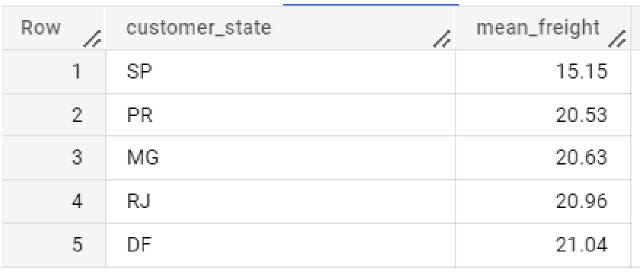
Row	customer_state	mean_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.15	8.66	11.21
2	PR	20.53	11.89	13.49
3	MG	20.63	11.92	13.34
4	RJ	20.96	15.07	12.01
5	DF	21.04	12.89	12.2
6	SC	21.47	14.95	11.57
7	RS	21.74	15.13	14.13
8	ES	22.06	15.59	10.65
9	GO	22.77	15.34	12.29
10	MS	23.37	15.46	11.23

4. Sort the data to get the following:

a. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Top 5 states with lowest avg freight values

```
SELECT

c.customer_state,

ROUND(AVG(ot.freight_value),2) AS mean_freight,

FROM `Business_Case.order_items` as ot

JOIN `Business_Case.orders` as o

ON ot.order_id=o.order_id

JOIN `Business_Case.customers` as c

ON o.customer_id=c.customer_id

GROUP BY c.customer_state

ORDER BY mean_freight

LIMIT 5
```

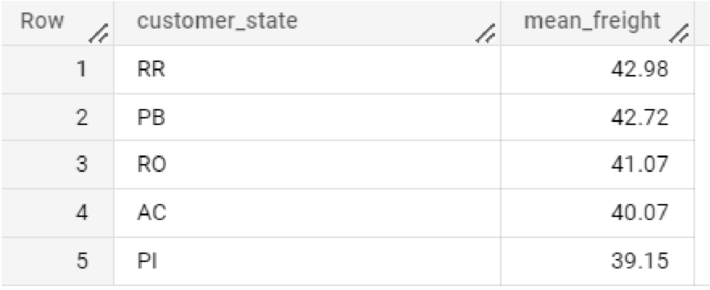


Row	customer_state	mean_freight
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

Top 5 states with highest avg freight values

```
SELECT

c.customer_state,

ROUND(AVG(ot.freight_value),2) AS mean_freight,

FROM `Business_Case.order_items` as ot

JOIN `Business_Case.orders` as o
```

```
ON ot.order_id=o.order_id

JOIN `Business_Case.customers` as c

ON o.customer_id=c.customer_id

GROUP BY c.customer_state

ORDER BY mean_freight DESC

LIMIT 5
```



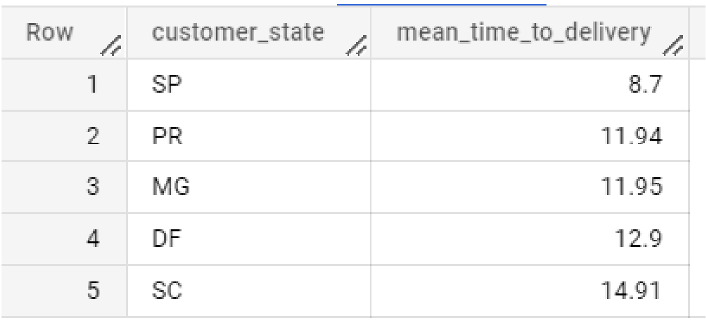
Row	customer_state	mean_freight
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

b. Top 5 states with highest/lowest average time to delivery

Top 5 states with lowest avg time to delivery

```
SELECT

c.customer_state,

ROUND(AVG(DATE_DIFF(t.delivery_date,t.purchase_date, DAY)),2) AS mean_time_to_delivery,

FROM

(SELECT

order_id,

customer_id,

EXTRACT(date FROM order_purchase_timestamp) as purchase_date,

EXTRACT(date FROM order_delivered_customer_date) as delivery_date,

FROM `Business_Case.orders`)AS t

JOIN `Business_Case.customers` as c

ON t.customer_id=c.customer_id
```

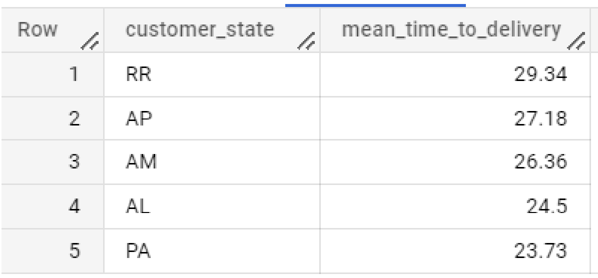
```
GROUP BY c.customer_state

ORDER BY mean_time_to_delivery

LIMIT 5
```



Row	customer_state	mean_time_to_delivery
1	SP	8.7
2	PR	11.94
3	MG	11.95
4	DF	12.9
5	SC	14.91

Top 5 states with highest avg time to delivery

```
SELECT

c.customer_state,

ROUND(AVG(DATE_DIFF(t.delivery_date,t.purchase_date, DAY)),2) AS mean_time_to_delivery,

FROM

(SELECT

order_id,

customer_id,

EXTRACT(date FROM order_purchase_timestamp) as purchase_date,

EXTRACT(date FROM order_delivered_customer_date) as delivery_date,

FROM `Business_Case.orders`)AS t

JOIN `Business_Case.customers` as c

ON t.customer_id=c.customer_id

GROUP BY c.customer_state

ORDER BY mean_time_to_delivery DESC

LIMIT 5
```

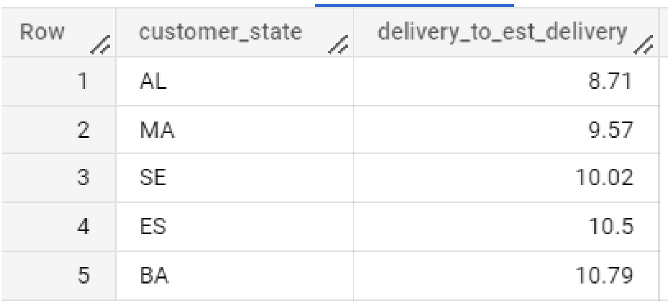


Row	customer_state	mean_time_to_delivery
1	RR	29.34
2	AP	27.18
3	AM	26.36
4	AL	24.5
5	PA	23.73

c. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Top 5 states where delivery is slow as compared to estimated date

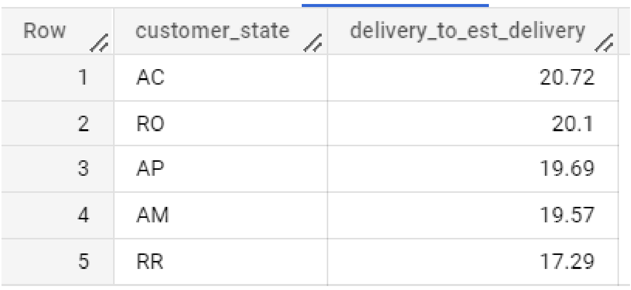
```
SELECT
c.customer_state,
ROUND(AVG(DATE_DIFF(t.est_delivery_date,t.delivery_date, DAY)),2) AS delivery_to_est_delivery,
FROM
(SELECT
order_id,
customer_id,
EXTRACT(date FROM order_purchase_timestamp) as purchase_date,
EXTRACT(date FROM order_delivered_customer_date) as delivery_date,
EXTRACT(date FROM order_estimated_delivery_date) as est_delivery_date
FROM `Business_Case.orders`)AS t
JOIN `Business_Case.customers` as c
ON t.customer_id=c.customer_id
GROUP BY c.customer_state
ORDER BY delivery_to_est_delivery
LIMIT 5
```



Row	customer_state	delivery_to_est_delivery
1	AL	8.71
2	MA	9.57
3	SE	10.02
4	ES	10.5
5	BA	10.79

Top 5 states where delivery is fast as compared to estimated date

```
SELECT

c.customer_state,

ROUND(AVG(DATE_DIFF(t.est_delivery_date,t.delivery_date, DAY)),2) AS delivery_to_est_delivery,

FROM

(SELECT

order_id,

customer_id,

EXTRACT(date FROM order_purchase_timestamp) as purchase_date,

EXTRACT(date FROM order_delivered_customer_date) as delivery_date,

EXTRACT(date FROM order_estimated_delivery_date) as est_delivery_date

FROM `Business_Case.orders`)AS t

JOIN `Business_Case.customers` as c

ON t.customer_id=c.customer_id

GROUP BY c.customer_state

ORDER BY delivery_to_est_delivery DESC

LIMIT 5
```



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

Insights:

- Mean freight is the lowerst in 'SP' state with the fastest delivery of the product. We can see that the company takes less time to deliver the product than the estitated time that helps reduce the freight cost.

Recommendation:

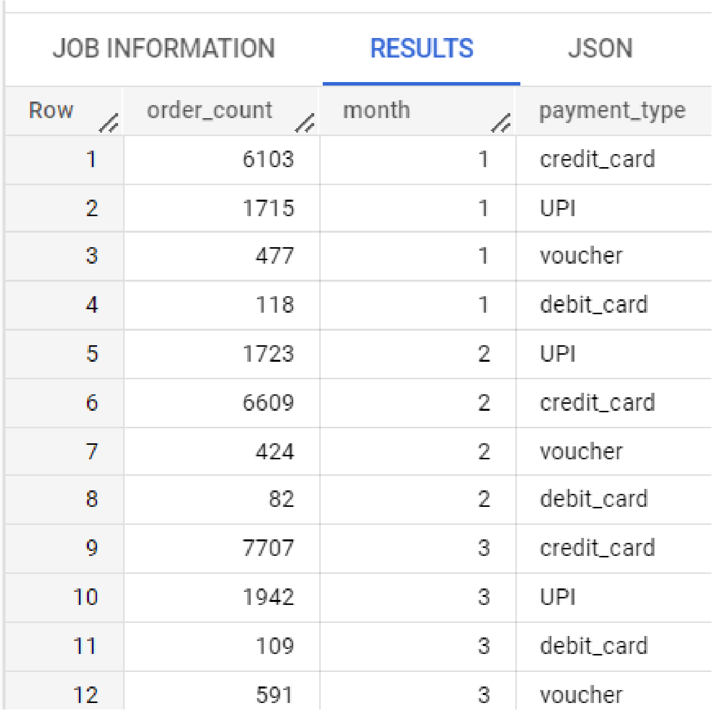
- We need to reduce the delivery time for the product so we can bring down the freight charges and enhance our services in other states like we have in ‘SP’.

F. Payment type analysis:

1. Month over Month count of orders for different payment types

```
SELECT

COUNT(t.order_id)as order_count,

t.month,

t.payment_type

FROM

  (SELECT

  o.order_id,

  EXTRACT(month FROM o.order_purchase_timestamp) as Month,

  p.payment_type

  FROM `Business_Case.orders` as o

  JOIN `Business_Case.payments` as p
```

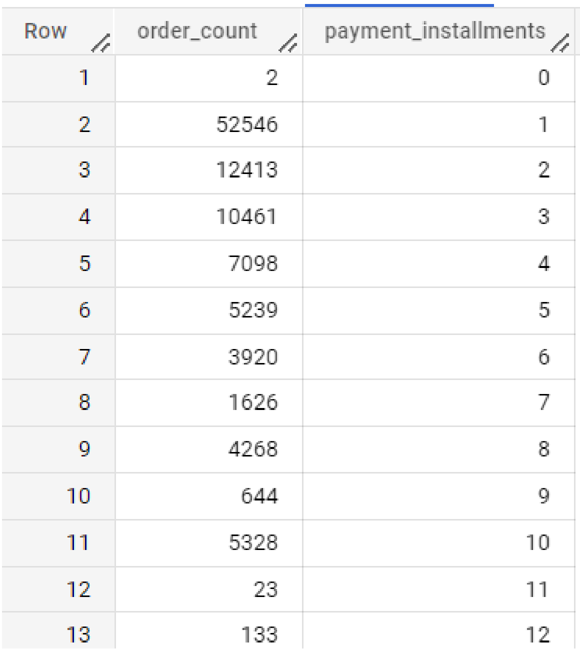
```
 ON o.order_id=p.order_id) as t

GROUP BY t.payment_type, t.month

ORDER BY t.month
```



Row	order_count	month	payment_type
1	6103	1	credit_card
2	1715	1	UPI
3	477	1	voucher
4	118	1	debit_card
5	1723	2	UPI
6	6609	2	credit_card
7	424	2	voucher
8	82	2	debit_card
9	7707	3	credit_card
10	1942	3	UPI
11	109	3	debit_card
12	591	3	voucher

2. Count of orders based on the no. of payment installments

```
SELECT

count(o.order_id) AS order_count,

p.payment_installments

FROM `Business_Case.orders` as o

JOIN `Business_Case.payments` as p

ON o.order_id=p.order_id

GROUP BY p.payment_installments

ORDER BY p.payment_installments
```



Row	order_count	payment_installments
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8
10	644	9
11	5328	10
12	23	11
13	133	12

Insights:

- There are 4 payment methods i.e., credit card, UPI, voucher, and debit card. However, we can see that a significant chunk of the customer chooses to pay with credit cards.
  .

Recommendation:

- We can add more payment options for customers to they have liberty on payment options.
- We can add more offers on debit cards and UPI payment options by collaborating with third parties or with banks.