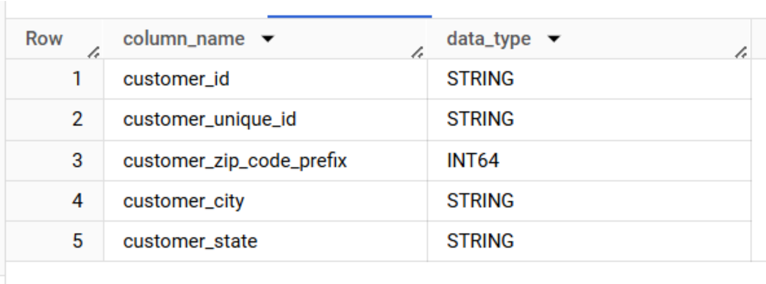
ANSWER SHEET

1. Initial Exploratory Analysis

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

```
select column_name, data_type
from `sal-dsml-sql.CaseStudy.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers'
```

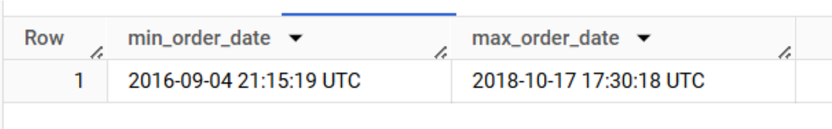


Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Note: this query gives the columns name and datatype of all the columns.

1.b. Get the time range between which the orders were placed.

```
select min(order_purchase_timestamp) as min_order_date,
      max(order_purchase_timestamp) as max_order_date
from `CaseStudy.orders`
```

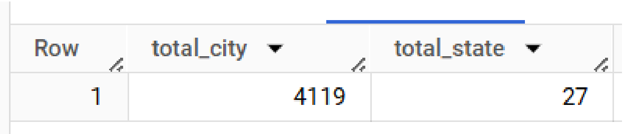


Row	min_order_date	max_order_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Note: this query present the date of the first order and the latest order date

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

```
select count(distinct customer_city) as total_city,
      count(distinct customer_state) as total_state
from  `CaseStudy.customers`
```



Row	total_city	total_state
1	4119	27

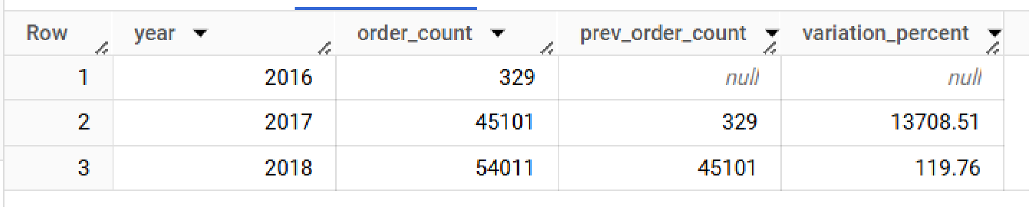
Note: the above query actually shows the number of cities which have been catered till now by the target company along with the total states it provides services to.

Summary : overall in the first section we are just getting the gist about the target company working in brazil like how many cities the order is coming and since when the company is functioning in brazil.

2. In-depth Exploration

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

```
select *, lag(order_count) over(order by year) prev_order_count,
        round(100*order_count/(lag(order_count) over(order by year)),2) as
variation_percentage
from(
SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
    COUNT(*) AS order_count
FROM `CaseStudy.orders`
GROUP BY year
)
order by year
```

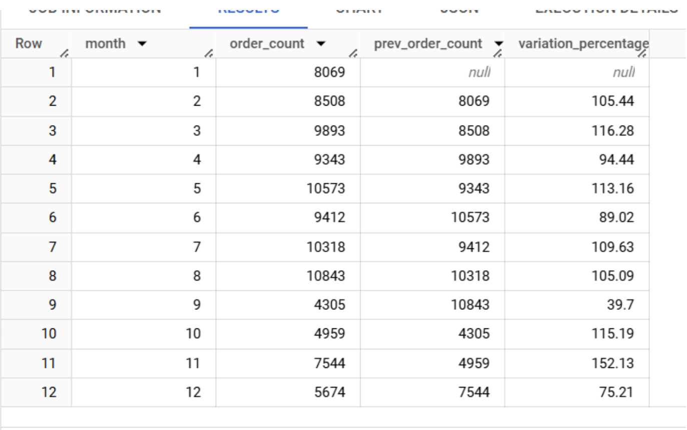


Row	year	order_count	prev_order_count	variation_percent
1	2016	329	*null*	*null*
2	2017	45101	329	13708.51
3	2018	54011	45101	119.76

Note : if variation percent is above hundred there is increase in total order else if it is less than 100 then there is a decrease.
Here we can see since the target was launched in 2016 in Brazil the pace it picked up was huge as compared to the following years.

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

```
select *, lag(order_count) over(order by month) prev_order_count,
        round(100*order_count/(lag(order_count) over(order by month)),2) as
variation_percentage
from(
SELECT
    extract(month from order_purchase_timestamp) AS month,
    COUNT(*) AS order_count
FROM `CaseStudy.orders`
GROUP BY month
)
order by month
```



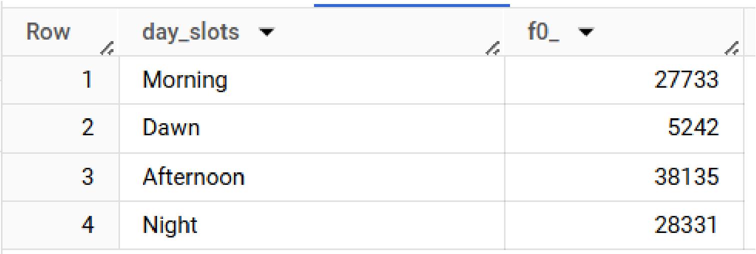
Row	month	order_count	prev_order_count	variation_percentage
1	1	8069	null	null
2	2	8508	8069	105.44
3	3	9893	8508	116.28
4	4	9343	9893	94.44
5	5	10573	9343	113.16
6	6	9412	10573	89.02
7	7	10318	9412	109.63
8	8	10843	10318	105.09
9	9	4305	10843	39.7
10	10	4959	4305	115.19
11	11	7544	4959	152.13
12	12	5674	7544	75.21

Note: here we are trying to check a pattern for which month the order was more compared to other months.

2.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 : 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
'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 day_slots, count (*)
from `CaseStudy.orders`
group by day_slots
```



Row	day_slots	f0_
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

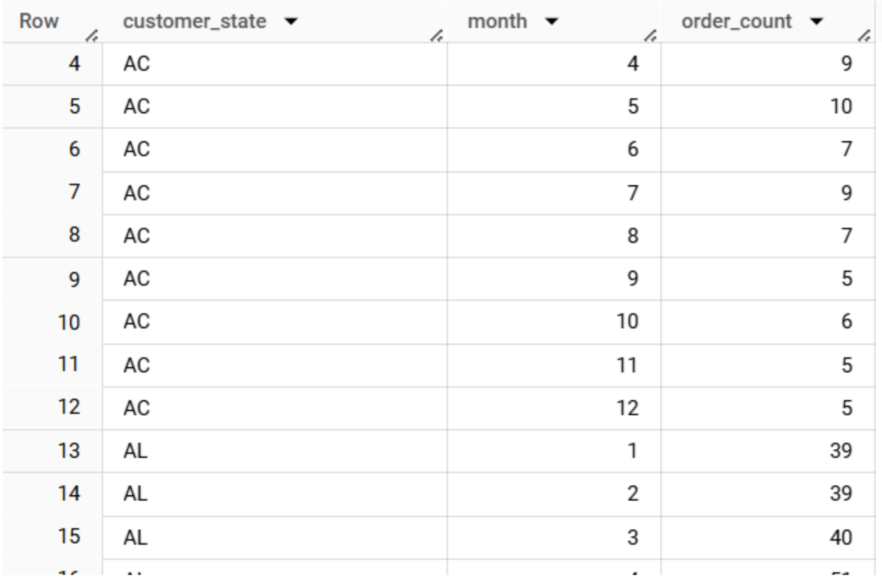
Note : we are getting to know that most people would prefer making an order in the afternoon.

Overall Summary : we are getting a sense of orders being made on a monthly basis and day wise basis. Along with the percentage change as compared with the previous month.

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

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

```
select c.customer_state, extract(month from o.order_purchase_timestamp) as month,
count(*) as order_count
from `CaseStudy.orders` o join `CaseStudy.customers` c using(customer_id)
group by c.customer_state, month
order by c.customer_state, month
```

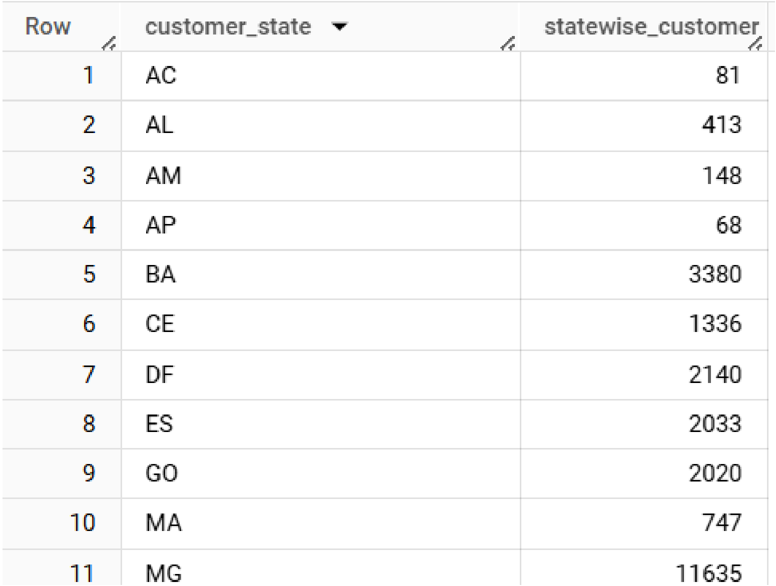


Row	customer_state	month	order_count
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5
12	AC	12	5
13	AL	1	39
14	AL	2	39
15	AL	3	40

Note: we can observe that every month from every state how many orders are getting placed.

3.b. How are the customers distributed across all the states?

```
select customer_state, count(*) as statewise_customer
from `CaseStudy.customers`
group by customer_state
order by customer_state
```



Row	customer_state	statewise_customer
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635

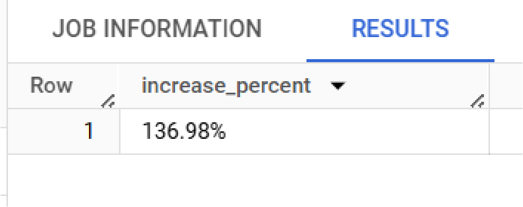
Note: here we are checking the state wise data alphabetical wise how many orders are being placed by each state.

Summary : In section 3 we are getting the data on orders on the regional basis.

4. Impact on Economy:

4.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.

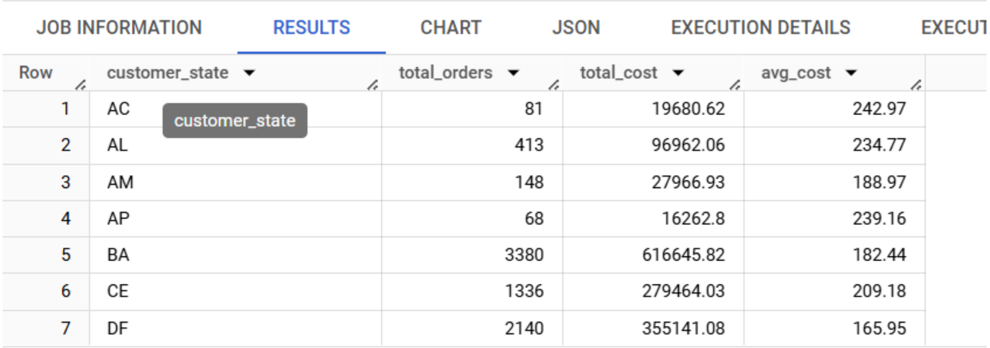
```
select concat(round(100*(order_cost-lead(order_cost) over(order by year desc))/
lead(order_cost) over(order by year desc),2),'%') as increase_percent
from
(select extract(year from o.order_purchase_timestamp) as year,
round(sum(p.payment_value),2) as order_cost
from `CaseStudy.orders` o join `CaseStudy.payments` p on o.order_id=p.order_id
where extract(year from o.order_purchase_timestamp) in (2017,2018) and
      extract(month from o.order_purchase_timestamp) between 0 and 8
group by 1) as tbl
order by 1 desc
limit 1
```



Row	increase_percent
1	136.98%

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

```
select c.customer_state, count(t.order_cost) as total_orders ,round(sum(t.order_cost),2)
as total_cost,round(avg(t.order_cost),2) as avg_cost
from `CaseStudy.customers` c join
(select o.customer_id as customer_id,
round(sum(p.payment_value),2) as order_cost
from `CaseStudy.orders` o join `CaseStudy.payments` p on o.order_id=p.order_id
group by 1) t on c.customer_id=t.customer_id
group by 1
order by 1
```

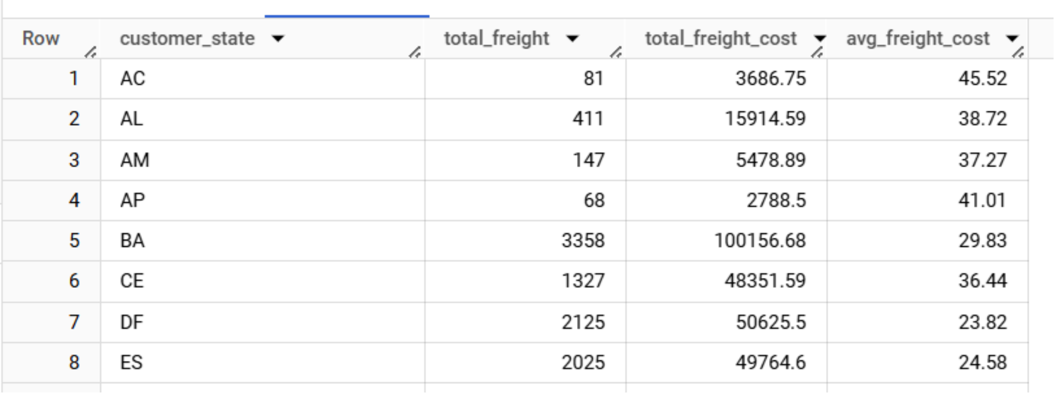


Row	customer_state	total_orders	total_cost	avg_cost
1	AC	81	19680.62	242.97
2	AL	413	96962.06	234.77
3	AM	148	27966.93	188.97
4	AP	68	16262.8	239.16
5	BA	3380	616645.82	182.44
6	CE	1336	279464.03	209.18
7	DF	2140	355141.08	165.95

Note: we can see from the data which state are having the huge orders revenue along with the avg price of all the orders in each state

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

```
select c.customer_state, count(t.freight_cost) as total_freight
,round(sum(t.freight_cost),2) as total_freight_cost,round(avg(t.freight_cost),2) as
avg_freight_cost
from `CaseStudy.customers` c join
(select o.customer_id as customer_id,
round(sum(p.freight_value),2) as freight_cost
from `CaseStudy.orders` o join `CaseStudy.order_items` p on o.order_id=p.order_id
group by 1) t on c.customer_id=t.customer_id
group by 1
order by 1
```



Row	customer_state	total_freight	total_freight_cost	avg_freight_cost
1	AC	81	3686.75	45.52
2	AL	411	15914.59	38.72
3	AM	147	5478.89	37.27
4	AP	68	2788.5	41.01
5	BA	3358	100156.68	29.83
6	CE	1327	48351.59	36.44
7	DF	2125	50625.5	23.82
8	ES	2025	49764.6	24.58

Summary: In sedition 4 we are trying to get a better insights on the cost of orders and the revenue each state is contributing for the target company.

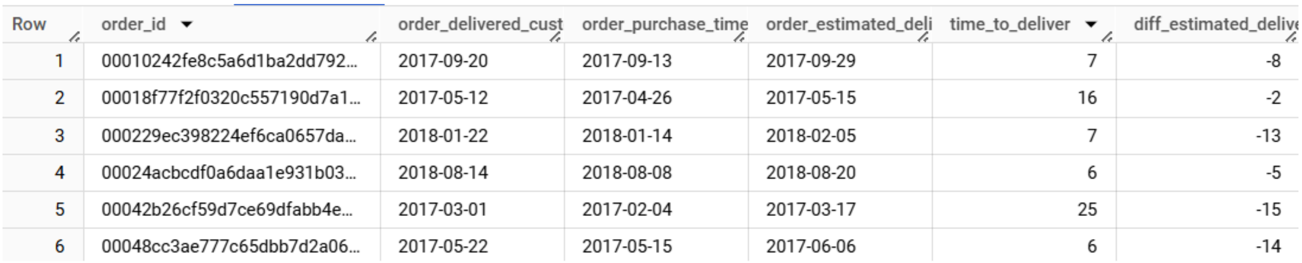
5. Analysis based on sales, freight and delivery time

5.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_delivered_customer_date - order_estimated_delivery_date

```
select order_id, date(order_delivered_customer_date) as order_delivered_customer_date,
             date(order_purchase_timestamp) as order_purchase_timestamp,
             date(order_estimated_delivery_date) as order_estimated_delivery_date,
      date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
time_to_deliver,
      date_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as
diff_estimated_delivery
from `CaseStudy.orders`
```

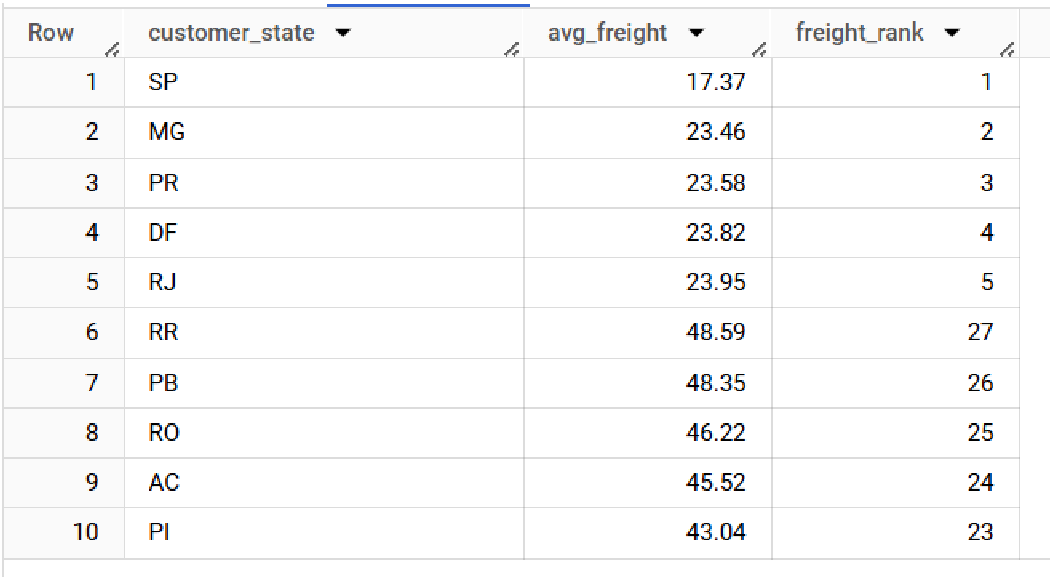


Row	order_id	order_delivered_cust	order_purchase_time	order_estimated_deli	time_to_deliver	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	2017-09-20	2017-09-13	2017-09-29	7	-8
2	00018f77f2f0320c557190d7a1...	2017-05-12	2017-04-26	2017-05-15	16	-2
3	000229ec398224ef6ca0657da...	2018-01-22	2018-01-14	2018-02-05	7	-13
4	00024acbcdf0a6daa1e931b03...	2018-08-14	2018-08-08	2018-08-20	6	-5
5	00042b26cf59d7ce69dfabb4e...	2017-03-01	2017-02-04	2017-03-17	25	-15
6	00048cc3ae777c65dbb7d2a06...	2017-05-22	2017-05-15	2017-06-06	6	-14

Note : positive diff_estimated_delivery value means the order arrived x days early, negative diff_estimated_delivery value means that it took that many days late to deliver the order.

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

```
with freight_orders as (
select c.customer_state,round(avg(ot.total_freight),2) avg_freight, dense_rank()
over(order by avg(ot.total_freight)) as freight_rank
from `CaseStudy.customers` c join `CaseStudy.orders` o
on c.customer_id=o.customer_id
join
(select order_id, round(sum(freight_value),2) as total_freight
from `CaseStudy.order_items`
group by 1) as ot on ot.order_id = o.order_id
group by 1
order by 2)
select * from
(select *
from freight_orders
order by freight_rank
limit 5)
union all
select * from
(select *
from freight_orders
order by freight_rank desc
limit 5)
```



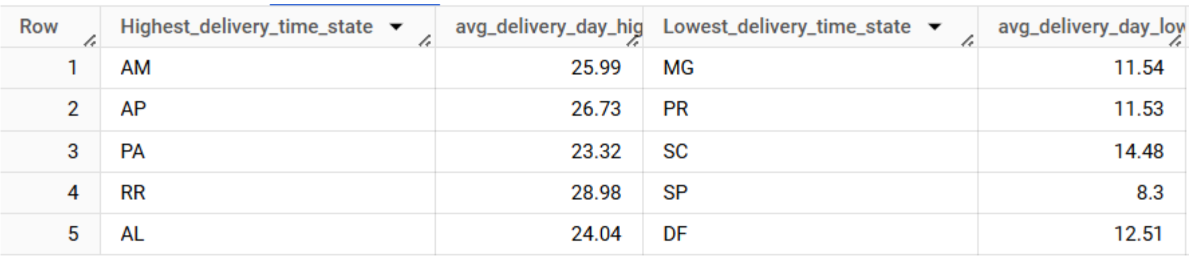
Row	customer_state	avg_freight	freight_rank
1	SP	17.37	1
2	MG	23.46	2
3	PR	23.58	3
4	DF	23.82	4
5	RJ	23.95	5
6	RR	48.59	27
7	PB	48.35	26
8	RO	46.22	25
9	AC	45.52	24
10	PI	43.04	23

Note: we can check that the avg freight value is ordered in ascending order and then the ranking is based. First 5 rows show the lowest avg_freight value and bottom 5 rows show the highest

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

```
with avg_delivery_time as (

select c.customer_state as state,
round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,
day)),2) as avg_delivery_day
from `CaseStudy.customers` c join `CaseStudy.orders` o on c.customer_id=o.customer_id
group by 1
order by 1
),
avg_delivery_rank as (
select *, dense_rank() over(order by avg_delivery_day) as L_rank,
            dense_rank() over(order by avg_delivery_day desc) as H_rank
from avg_delivery_time
order by L_rank
)
select h.state as Highest_delivery_time_state, h.avg_delivery_day as
avg_delivery_day_high,
l.state as Lowest_delivery_time_state, l.avg_delivery_day as avg_delivery_day_low
from avg_delivery_rank h join avg_delivery_rank l on h.H_rank=l.L_rank
where h.H_rank<=5
```

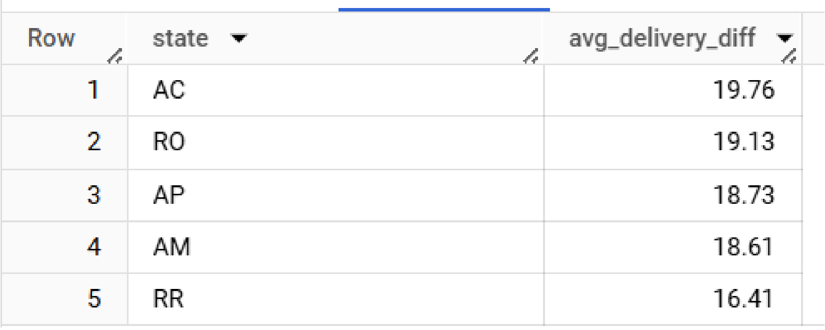


Row	Highest_delivery_time_state	avg_delivery_day_hig	Lowest_delivery_time_state	avg_delivery_day_low
1	AM	25.99	MG	11.54
2	AP	26.73	PR	11.53
3	PA	23.32	SC	14.48
4	RR	28.98	SP	8.3
5	AL	24.04	DF	12.51

Note: i am trying to approach the problem with different solutions of showing top5 and bottom 5 data.

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.

```
with cte as (
select c.customer_state as state,
round(avg(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date,
day)),2) as avg_delivery_diff,
from `CaseStudy.customers` c join `CaseStudy.orders` o on c.customer_id=o.customer_id
group by 1
order by avg_delivery_diff desc
)
select * from cte
limit 5
```



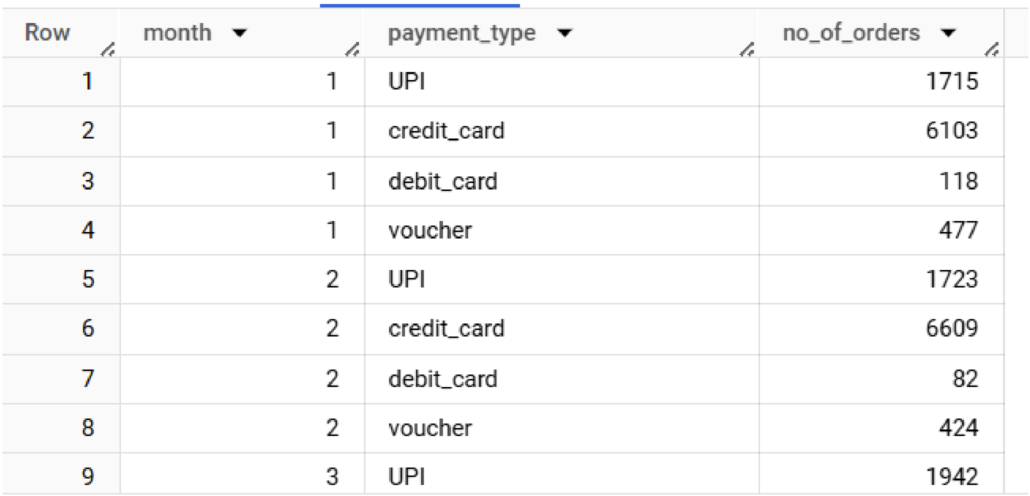
Row	state	avg_delivery_diff
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Note: As we can see that the more delivery day difference is there the faster the order is delivered.

6. Analysis based on the payments

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

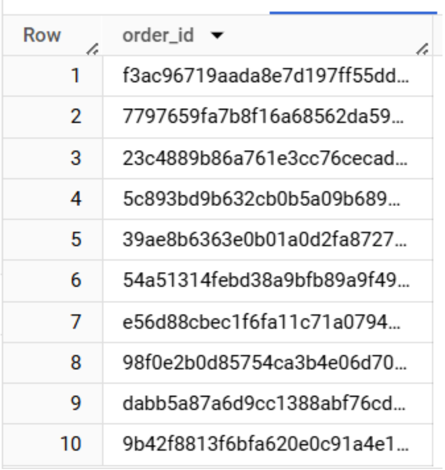
```
select extract(month FROM o.order_purchase_timestamp) as month,p.payment_type, count(*)
as no_of_orders
from `CaseStudy.payments` p join `CaseStudy.orders` o on o.order_id=p.order_id
group by 1,2
order by month, p.payment_type
```



Row	month	payment_type	no_of_orders
1	1	UPI	1715
2	1	credit_card	6103
3	1	debit_card	118
4	1	voucher	477
5	2	UPI	1723
6	2	credit_card	6609
7	2	debit_card	82
8	2	voucher	424
9	3	UPI	1942

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

```
select distinct order_id
from `CaseStudy.payments`
where payment_type = 'credit_card'
and payment_installments > 1
group by 1
having count(*) > 1
```



Row	order_id
1	f3ac96719aada8e7d197ff55dd...
2	7797659fa7b8f16a68562da59...
3	23c4889b86a761e3cc76cecad...
4	5c893bd9b632cb0b5a09b689...
5	39ae8b6363e0b01a0d2fa8727...
6	54a51314febd38a9bfb89a9f49...
7	e56d88cbec1f6fa11c71a0794...
8	98f0e2b0d85754ca3b4e06d70...
9	dabb5a87a6d9cc1388abf76cd...
10	9b42f8813f6bfa620e0c91a4e1...

Note : We got the orders on installment where at least 1 emi has been paid.

Few Important Analysis based on the query executed for Target:

- As per 2c order made on dawn is comparatively less so all the maintenance and deployment testing should be done at that time.
- As per 2b sale during the month of sept to dec is comparatively less as compared to jan - aug. so the best time for marketing is during the month of aug in order to increase the sale throughout the year.
- As per 3b & 4b the state having less than 1000 customers or less than 1000 orders should be given more focus on marketing in those areas.
- On the basis of 5a, Target should focus on opening more inventory in areas where delivery is late.
- On the basis of the 5th segment, Target should not take more than 15 days for any item to be delivered. It will build more trust and reliability in view of customers.

Delivered by - **Salsabil Ahmad**

(Target SQL Case study of Scalar)