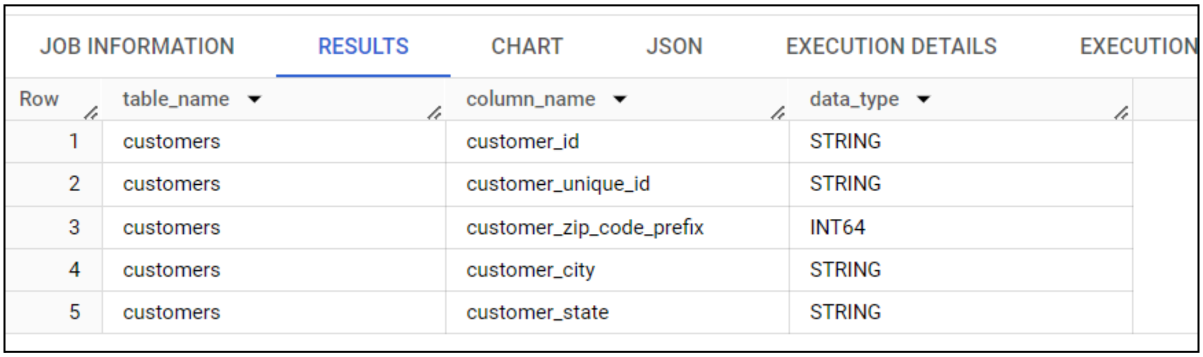
CASE STUDY - TARGET

1.

A.

```
SELECT table_name, column_name, data_type
FROM Target_Dataset.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
```

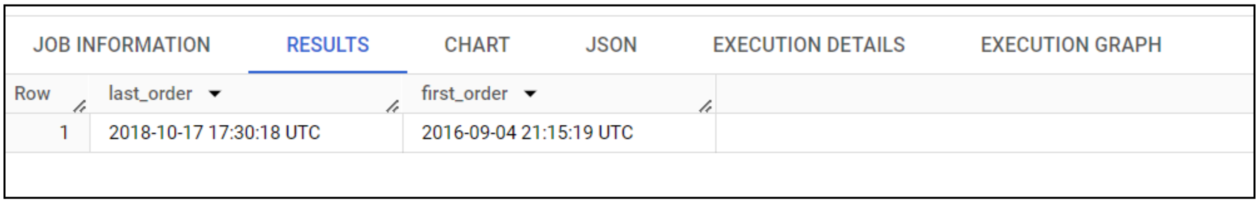


Row	table_name	column_name	data_type
1	customers	customer_id	STRING
2	customers	customer_unique_id	STRING
3	customers	customer_zip_code_prefix	INT64
4	customers	customer_city	STRING
5	customers	customer_state	STRING

Inference: We need to select from the information schema to get the schema (data_type) of the columns of a certain table.

B.

```
SELECT max(order_purchase_timestamp) as last_order,
min(order_purchase_timestamp) as first_order
FROM `case-study-1-430318.Target_Dataset.orders` LIMIT 1000
```



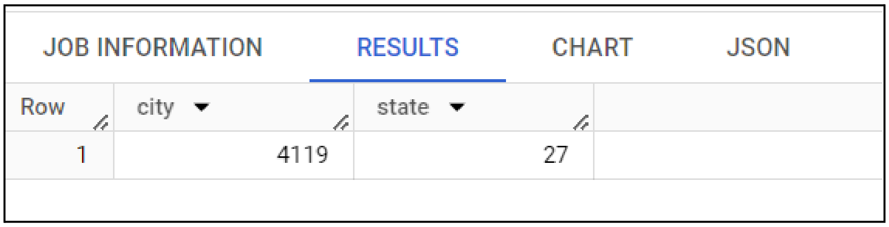
Row	last_order	first_order
1	2018-10-17 17:30:18 UTC	2016-09-04 21:15:19 UTC

Inference: We need to find the range of the first and last order. That's why I selected max and min values from the order_purchase_timestamp column in the orders table. The first order was in 2016 and last was in 2018.

C.

```
SELECT
count(distinct c.customer_city) as city,
count(distinct c.customer_state) as state
```

```
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
```



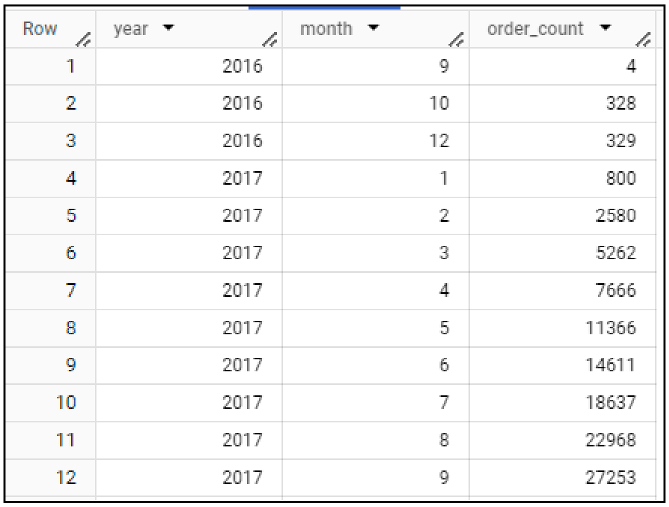
Row	city	state
1	4119	27

Inference: To calculate the total number of city and state, I used count and it is observed that over the period 4119 is the total city count and 27 is the total state count.

2.

A.

```
select distinct year, month,
count(order_id) over (partition by year order by month asc) as order_count,
from
(SELECT extract(year from order_purchase_timestamp) as year,
order_id,
extract(month from order_purchase_timestamp) as month
 FROM `case-study-1-430318.Target_Dataset.orders`
) X
```

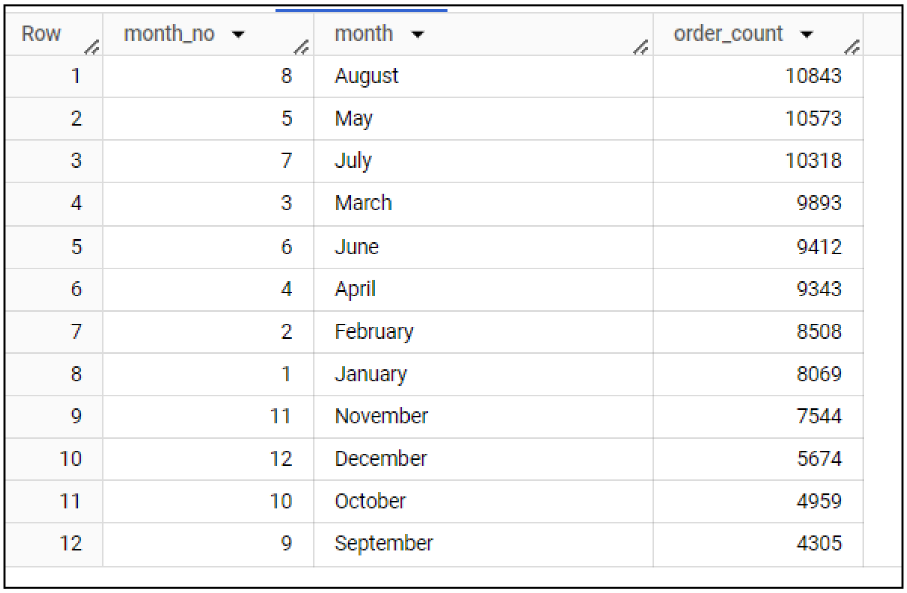


Row	year	month	order_count
1	2016	9	4
2	2016	10	328
3	2016	12	329
4	2017	1	800
5	2017	2	2580
6	2017	3	5262
7	2017	4	7666
8	2017	5	11366
9	2017	6	14611
10	2017	7	18637
11	2017	8	22968
12	2017	9	27253

Inference: In this first we need to sort the data according to each year and then all the 12 months in that year and then I took the count of order. As you can see in 2016, total count is 329 and in 2017 it is 45101, so there is gradual increase in the count number.

B.

```
select distinct month_no, month,
count(order_id) over (partition by month_no order by month_no asc) as order_count
from
(SELECT order_id,
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month_no,
FORMAT_DATE("%B", order_purchase_timestamp) AS month,
 FROM `case-study-1-430318.Target_Dataset.orders`
) X
order by order_count desc
```

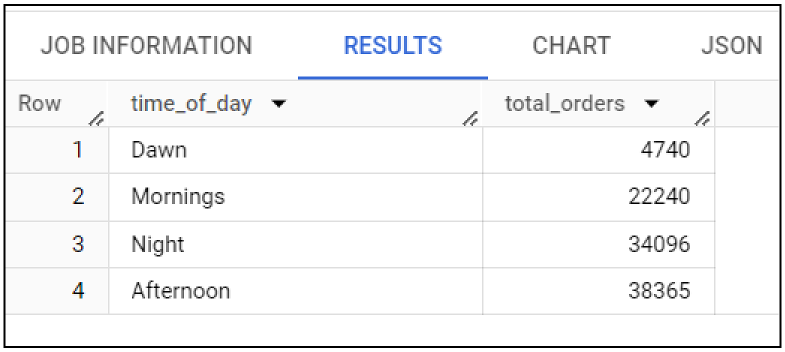


Row	month_no	month	order_count
1	8	August	10843
2	5	May	10573
3	7	July	10318
4	3	March	9893
5	6	June	9412
6	4	April	9343
7	2	February	8508
8	1	January	8069
9	11	November	7544
10	12	December	5674
11	10	October	4959
12	9	September	4305

Inference: I have counted the no of orders according to each month and it is observed that in August month,no. of orders placed are at its peak.

C.

```
Select time_of_day,
count(distinct order_id) as total_orders
from
(SELECT c.customer_id,o.order_id,
CAST(order_purchase_timestamp AS TIME) AS time,
(case when CAST(order_purchase_timestamp AS TIME) between '00:00:00' and '06:00:00'
then 'Dawn'
when CAST(order_purchase_timestamp AS TIME) between '06:00:00' and '12:00:00'
then 'Mornings'
when CAST(order_purchase_timestamp AS TIME) between '12:00:00' and '18:00:00'
then 'Afternoon'
when CAST(order_purchase_timestamp AS TIME) between '18:00:00' and '23:59:59'
then 'Night'
end ) as time_of_day
FROM `case-study-1-430318.Target_Dataset.customers` c join
`case-study-1-430318.Target_Dataset.orders` o
on c.customer_id = o.customer_id
) X
group by time_of_day
ORDER BY total_orders asc
```



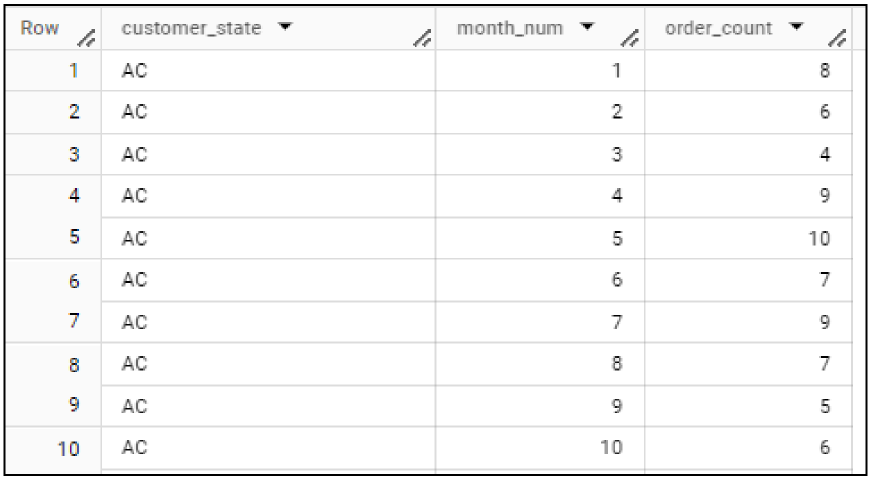
Row	time_of_day	total_orders
1	Dawn	4740
2	Mornings	22240
3	Night	34096
4	Afternoon	38365

Inference: I have grouped the given time in 4 stages: dawn, morning, afternoon and night. It is observed that at Afternoon the Brazilian customers usually order the most.

3.

A.

```
    Select customer_state, month_num, count(order_id) as order_count
    from
    (SELECT c.customer_id,c.customer_state,o.order_id,
    extract(month from o.order_purchase_timestamp) as month_num,
    format_date('%B', o.order_purchase_timestamp) as month
    FROM `case-study-1-430318.Target_Dataset.customers` c
    join
    `case-study-1-430318.Target_Dataset.orders` o
    on
    c.customer_id = o.customer_id
    ) X
    group by customer_state, month_num
    order by customer_state asc, month_num
```

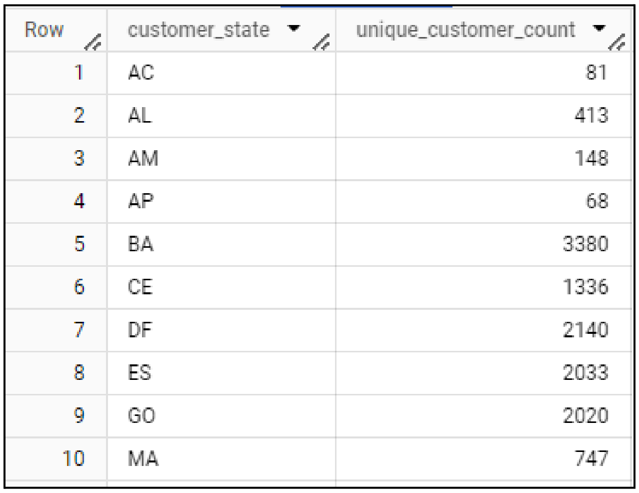


Row	customer_state	month_num	order_count
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Inference: I have grouped the data according to each state, in each month of the customers and have counted no. of orders at each unique time.

B.

```
Select customer_state,
count(distinct customer_id) as unique_customer_count
from
(SELECT c.customer_id,c.customer_state,o.order_id,
FROM `case-study-1-430318.Target_Dataset.customers` c join
`case-study-1-430318.Target_Dataset.orders` o
on c.customer_id = o.customer_id
) X
group by customer_state
order by customer_state asc
```



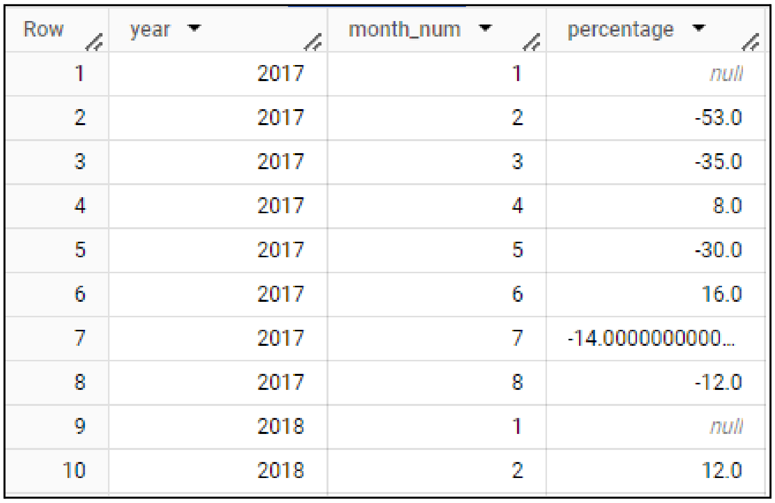
Row	customer_state	unique_customer_count
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

Inference: I have grouped the unique customer states and have counted unique no of customers in each state. With my observation MG is having the most customers as 11635 count and AP is having the least customers as 68.

4.

A.

```
select year,month_num,
ROUND((( next_month_cost - cost) / cost), 2) * 100 AS percentage
from
(select year, month_num, cost,
lag(cost) over (partition by year order by month_num asc) as next_month_cost
from
(SELECT
extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month_num,
SUM(p.payment_value) AS cost
FROM `case-study-1-430318.Target_Dataset.payments` p join
`case-study-1-430318.Target_Dataset.orders` o
on p.order_id =o.order_id
where (extract(month from o.order_purchase_timestamp) between 0 and 8) and
(extract(year from o.order_purchase_timestamp)) between 2017 and 2018
group by year, month_num
order by year, month_num asc) X) Y
order by year, month_num asc
```

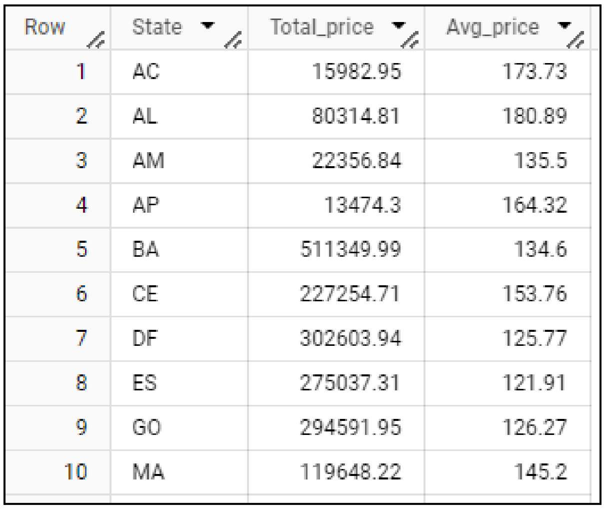


Row	year	month_num	percentage
1	2017	1	null
2	2017	2	-53.0
3	2017	3	-35.0
4	2017	4	8.0
5	2017	5	-30.0
6	2017	6	16.0
7	2017	7	-14.0000000000...
8	2017	8	-12.0
9	2018	1	null
10	2018	2	12.0

Inference: First we have to group the data with year then month and then we have to take the percentage increase of the cost of the orders by using lag function.
Example, in 2017, At June month we can see there is a percentage increase of 16 from year 2017 to 2018.

B.

```
select customer_state as State,
round(sum(price),2) as Total_price,
round(avg(price),2) as Avg_price
from
(SELECT i.order_id,c.customer_state,i.price
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
join
`case-study-1-430318.Target_Dataset.order_items` i
on i.order_id = o.order_id
order by c.customer_state asc) X
group by customer_state
order by customer_state asc
```

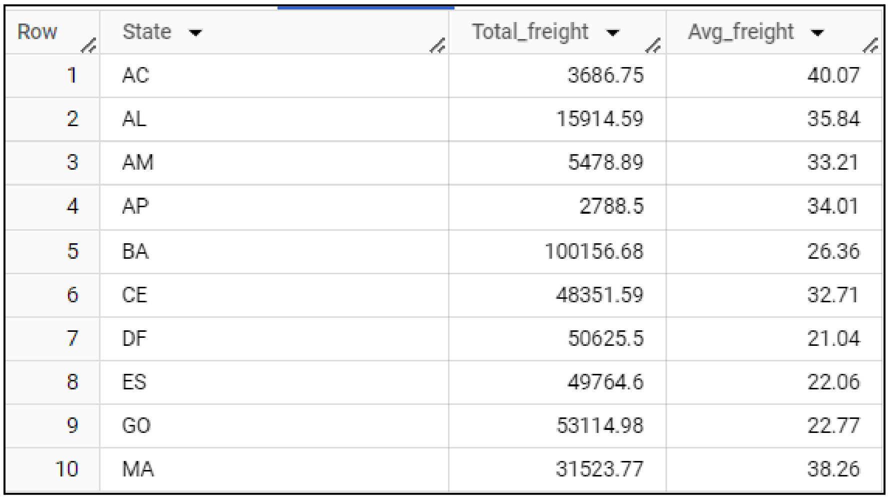


Row	State	Total_price	Avg_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

Inference: I have fetched the data by taking the distinct states first and then calculating the total_price by using sum function and avg_price by using the average function.

C.

```
select customer_state as State,
round(sum(freight_value),2) as Total_freight,
round(avg(freight_value),2) as Avg_freight
from
(SELECT i.order_id,c.customer_state,i.freight_value
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
join
`case-study-1-430318.Target_Dataset.order_items` i
on i.order_id = o.order_id
order by c.customer_state asc) X
group by customer_state
order by customer_state asc
```



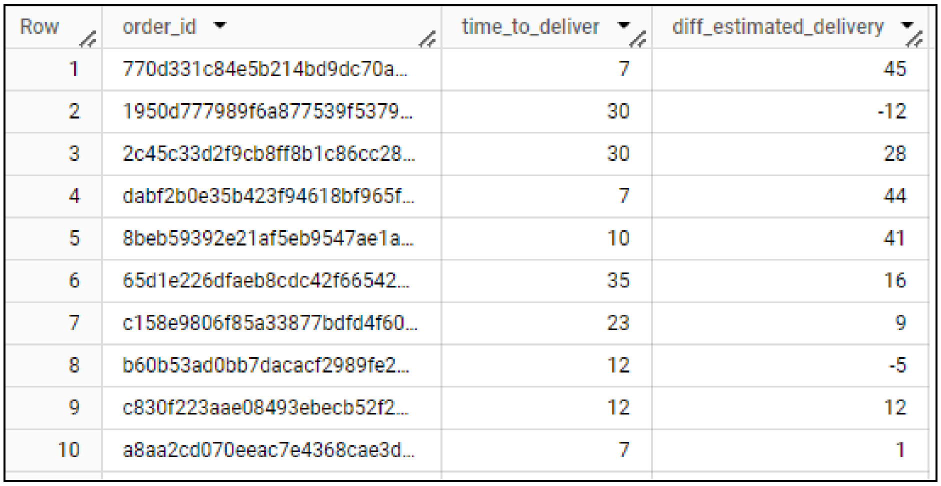
Row	State	Total_freight	Avg_freight
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

Inference: I have fetched the data by taking the distinct states first and then calculating the total_freight by using sum function and avg_freight by using the average function.

5.

A.

```
select *
from
(SELECT
Order_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM `case-study-1-430318.Target_Dataset.orders` ) X
where time_to_deliver is not NULL and diff_estimated_delivery is not NULL
limit 15
```



Row	order_id	time_to_deliver	diff_estimated_delivery
1	770d331c84e5b214bd9dc70a...	7	45
2	1950d777989f6a877539f5379...	30	-12
3	2c45c33d2f9cb8ff8b1c86cc28...	30	28
4	dabf2b0e35b423f94618bf965f...	7	44
5	8beb59392e21af5eb9547ae1a...	10	41
6	65d1e226dfaeb8cdc42f66542...	35	16
7	c158e9806f85a33877bdfd4f60...	23	9
8	b60b53ad0bb7dacacf2989fe2...	12	-5
9	c830f223aae08493ebecb52f2...	12	12
10	a8aa2cd070eeac7e4368cae3d...	7	1

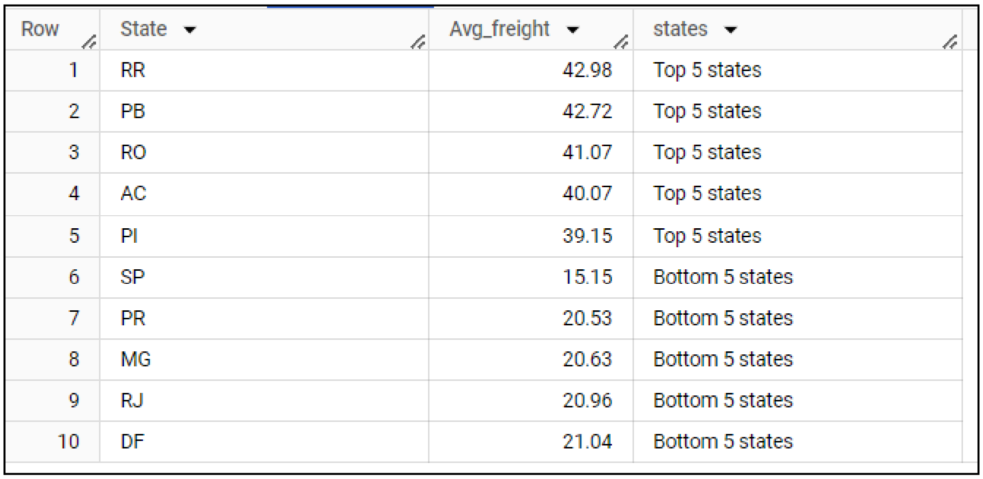
Inference: On the basis of order_id, i have calculated the delivery time and the difference between the estimated & actual delivery date. For row1, the time_to_deliver is 7 days but the difference in estimated delivery date is 45 days.

B.

```
(select customer_state as State,
round(avg(freight_value),2) as Avg_freight,
'Bottom 5 states' AS states
from
(SELECT i.order_id,c.customer_state,i.freight_value
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
join
`case-study-1-430318.Target_Dataset.order_items` i
on i.order_id = o.order_id
order by c.customer_state asc) X
group by customer_state
order by Avg_freight asc
limit 5)

UNION all

(select customer_state as State,
round(avg(freight_value),2) as Avg_freight,
'Top 5 states' AS states
from
(SELECT i.order_id,c.customer_state,i.freight_value
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
join
`case-study-1-430318.Target_Dataset.order_items` i
on i.order_id = o.order_id
order by c.customer_state asc) X
group by customer_state
order by Avg_freight desc, customer_state desc
limit 5 )
```



Row	State	Avg_freight	states
1	RR	42.98	Top 5 states
2	PB	42.72	Top 5 states
3	RO	41.07	Top 5 states
4	AC	40.07	Top 5 states
5	PI	39.15	Top 5 states
6	SP	15.15	Bottom 5 states
7	PR	20.53	Bottom 5 states
8	MG	20.63	Bottom 5 states
9	RJ	20.96	Bottom 5 states
10	DF	21.04	Bottom 5 states

Inference: I have simply taken the union all of the two queries. First one tells us about the Top 5 states and the second one tells us about the bottom 5 states on the basis of their Average Freight.

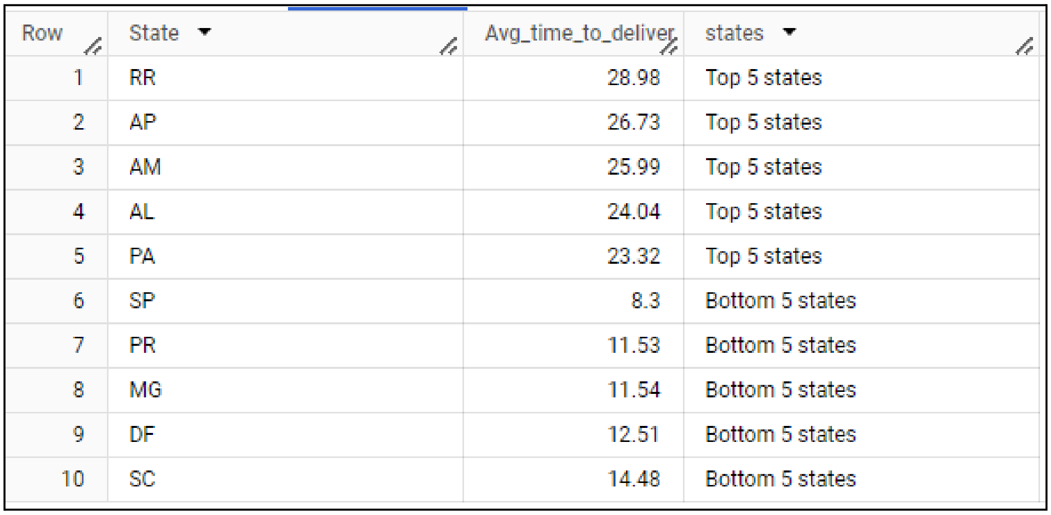
C.

```
    (select customer_state as State,
    round(avg(time_to_deliver),2) as Avg_time_to_deliver,
    'Bottom 5 states' AS states
    from
    (SELECT o.order_id,c.customer_state,
    DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
    time_to_deliver
    FROM `case-study-1-430318.Target_Dataset.orders` o join
    `case-study-1-430318.Target_Dataset.customers` c
    on o.customer_id = c.customer_id
    order by c.customer_state asc) X
```

```
group by customer_state
having Avg_time_to_deliver is not NULL
order by Avg_time_to_deliver asc
limit 5)

union all

(select customer_state as State,
round(avg(time_to_deliver),2) as Avg_time_to_deliver,
'Top 5 states' AS states
from
(SELECT o.order_id,c.customer_state,
DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_deliver
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
order by c.customer_state asc) X
group by customer_state
having Avg_time_to_deliver is not NULL
order by Avg_time_to_deliver desc
limit 5)
```

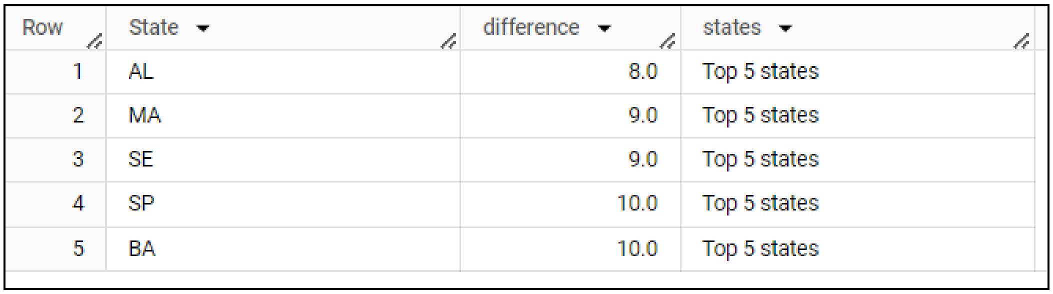


Row	State	Avg_time_to_deliver	states
1	RR	28.98	Top 5 states
2	AP	26.73	Top 5 states
3	AM	25.99	Top 5 states
4	AL	24.04	Top 5 states
5	PA	23.32	Top 5 states
6	SP	8.3	Bottom 5 states
7	PR	11.53	Bottom 5 states
8	MG	11.54	Bottom 5 states
9	DF	12.51	Bottom 5 states
10	SC	14.48	Bottom 5 states

Inference: I have simply taken the union all of the two queries. First one tells us about the Top 5 states and the second one tells us about the bottom 5 states on the basis of their average delivery time.

D.

```
select customer_state as State,
round(avg(fast_delivery)) as difference,
'Top 5 states' AS states
from
(SELECT o.order_id,c.customer_state,
DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) AS
fast_delivery
FROM `case-study-1-430318.Target_Dataset.orders` o join
`case-study-1-430318.Target_Dataset.customers` c
on o.customer_id = c.customer_id
order by fast_delivery asc) X
group by customer_state
having difference is not NULL
order by difference asc
limit 5
```



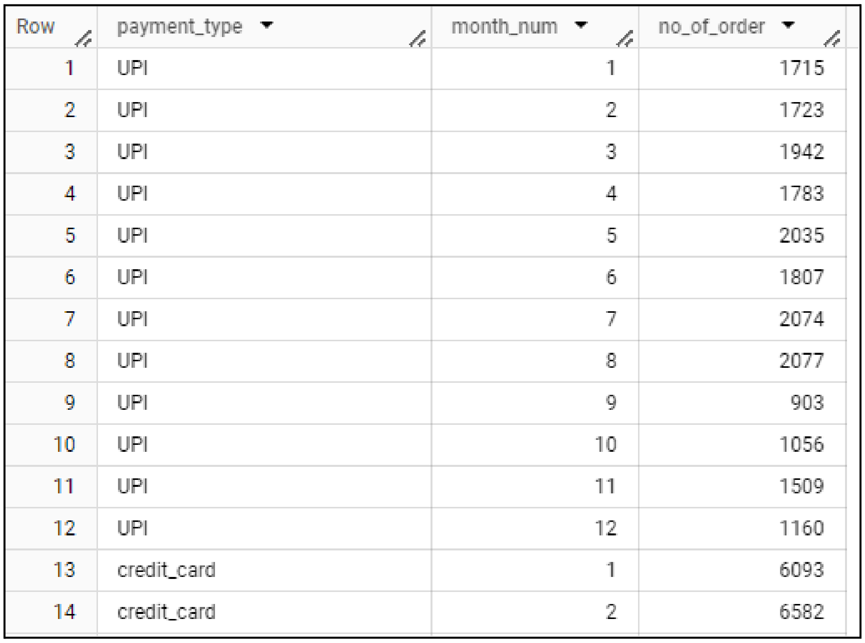
Row	State	difference	states
1	AL	8.0	Top 5 states
2	MA	9.0	Top 5 states
3	SE	9.0	Top 5 states
4	SP	10.0	Top 5 states
5	BA	10.0	Top 5 states

Inference: I have grouped the data by its state name and after that by using the function date_diff, i got to know the difference between the averages of actual & estimated delivery date which tells us how fast the delivery was for each state Then by using Limit function i extracted the top 5 data which tells us where the order delivery is really fast as compared to the estimated date of delivery.

6.

A.

```
select payment_type,month_num, c
ount(distinct order_id) as no_of_order
from
(SELECT o.order_id, p.payment_type,
extract(month from o.order_purchase_timestamp) as month_num,
format_date('%B',o.order_purchase_timestamp) as month
FROM `case-study-1-430318.Target_Dataset.payments` p join
`case-study-1-430318.Target_Dataset.orders` o
on p.order_id =o.order_id) X
group by payment_type,month_num
order by payment_type,month_num
```

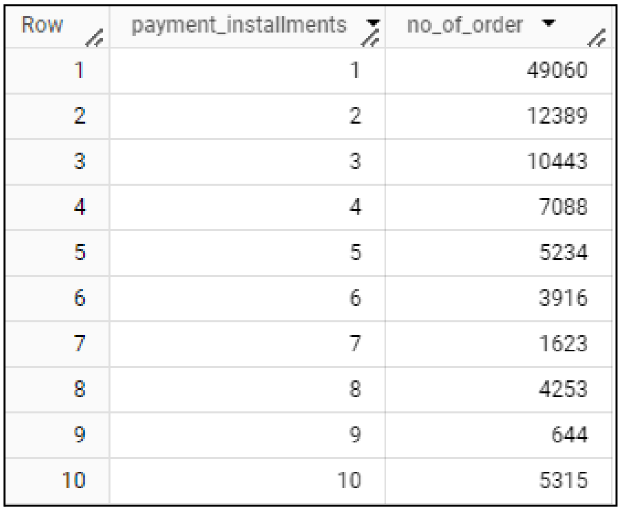


Row	payment_type	month_num	no_of_order
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6093
14	credit_card	2	6582

Inference: I have counted the no. of orders placed using the unique payment_type in each month over the past years. Example, UPI payment_type will have 1-12 months data of no_of_order same for all the other paymenttype.

B.

```
select payment_installments,
count(distinct order_id) as no_of_order
from
(SELECT o.order_id, p.payment_installments
FROM
`case-study-1-430318.Target_Dataset.payments` p
join
`case-study-1-430318.Target_Dataset.orders` o
on
p.order_id =o.order_id
where p.payment_installments >= 1)
group by payment_installments
order by payment_installments asc
```



Row	payment_installments	no_of_order
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315

Inference: I have counted the no. of orders placed based on the payment installments where at least one installment has been successfully paid. I have grouped the data by the payment_installment column and then counted the no of orders.