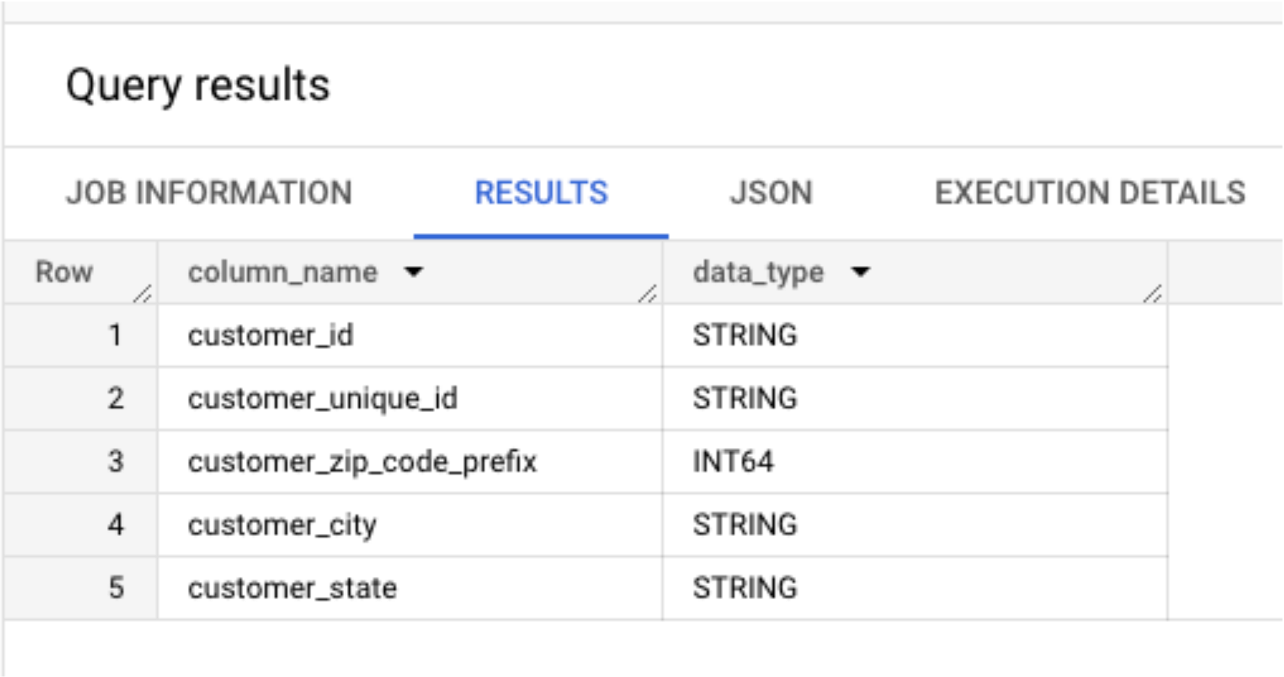
Question 1 - Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

1.1 - Data type of all columns in the "customers" table.

Answer :

```
SELECT
     column_name, data_type
FROM
`dsml-june23-sql`.target_store.INFORMATION_SCHEMA.COLUMNS
WHERE
 table_name = ‘customer’;
```

Result :



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

Insights :

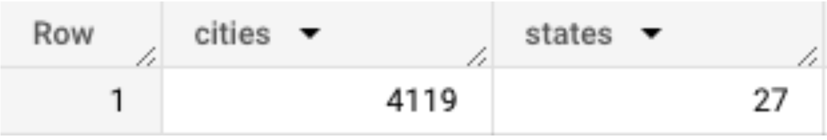
Above result shows data type of all the columns of customer table.

1.2 - Get the time range between which the orders were placed.

Answer :

```
SELECT
MIN(order_purchase_timestamp) as start_date,
MAX(order_purchase_timestamp) as end_date
FROM `target_store.orders`;
```

Result: :



Row	cities	states
1	4119	27

Insights :

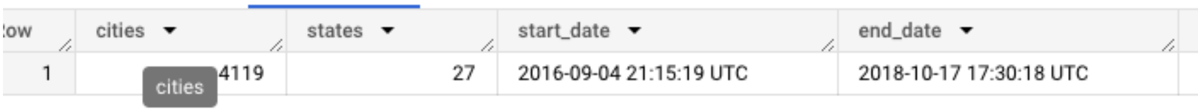
Above result shows that Orders getting placed from 04.09.2016 @ 09:15 PM and the last date when Target received the last order in Brazil is 17-10-2018 @ 05:30 pm.

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

Answer :

```
SELECT
COUNT(distinct c.customer_city) as cities,
COUNT(distinct c.customer_state) as states,
MIN(order_purchase_timestamp) as start_date,
MAX(order_purchase_timestamp) as end_date
FROM `target_store.orders` as o
JOIN  `target_store.customer` as c
ON c.customer_id = o.customer_id;
```

Result :



ow	cities	states	start_date	end_date
1	4119	27	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights :

From the start date of receiving orders i.e. 04-09-2016 and till the last date i.e. 17-10-2018,
Total cities from which orders received are 4119 and total states are 27.

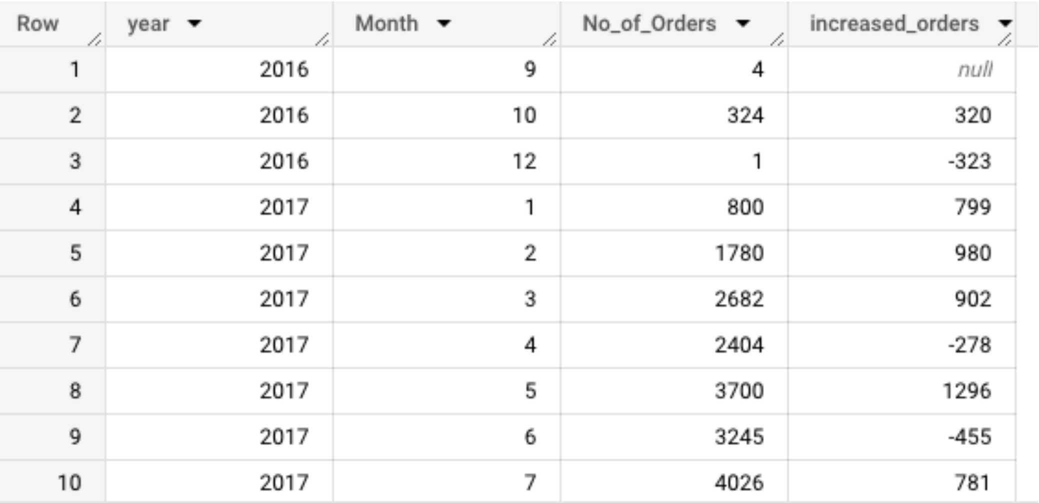
Question 2 - In-depth Exploration :

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

Answer :

```
Select * ,
No_of_Orders -
lag(No_of_orders)over(order by year, Month)  as increased_orders
from
(select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as Month,
count(order_id) as No_of_Orders
from `target_store.orders`
group by Month, year
) aa
order by year, Month
```

Result :



Row	year	Month	No_of_Orders	increased_orders
1	2016	9	4	*null*
2	2016	10	324	320
3	2016	12	1	-323
4	2017	1	800	799
5	2017	2	1780	980
6	2017	3	2682	902
7	2017	4	2404	-278
8	2017	5	3700	1296
9	2017	6	3245	-455
10	2017	7	4026	781

Insights :

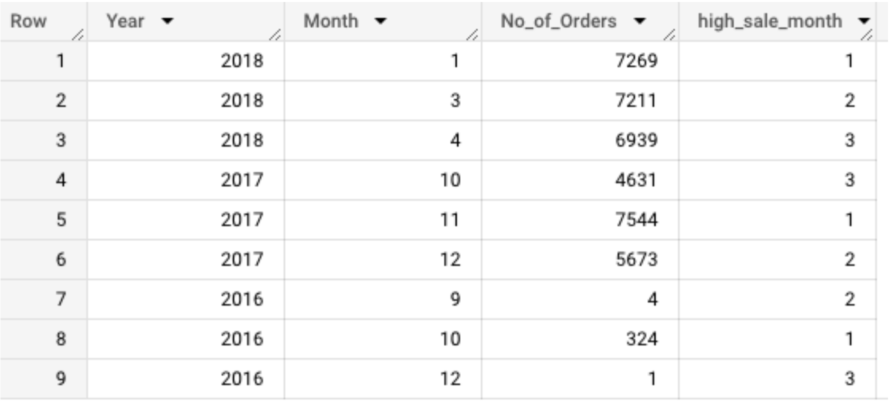
From above result we can conclude that the sale of Target is increased gradually in months over the part 3 year.
In 2016, total count of orders was 329.
In 2017, total count of orders was 45101.
In 2018, total count of orders was 54011.

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

Answer (a):

```
select Year, Month, No_of_Orders, high_sale_month from
(Select Year, Month, No_of_Orders,
dense_rank() over (partition by year order by No_of_Orders desc) as
high_sale_month
from
(select extract(year from order_purchase_timestamp) as Year,
extract (month from order_purchase_timestamp) as Month,
count(order_id) as No_of_Orders
from `target_store.orders`
group by Year, Month) aa
order by No_of_Orders desc) bb
where high_sale_month IN (1,2,3)
order by Year desc, Month
```

Result :



Row	Year	Month	No_of_Orders	high_sale_month
1	2018	1	7269	1
2	2018	3	7211	2
3	2018	4	6939	3
4	2017	10	4631	3
5	2017	11	7544	1
6	2017	12	5673	2
7	2016	9	4	2
8	2016	10	324	1
9	2016	12	1	3

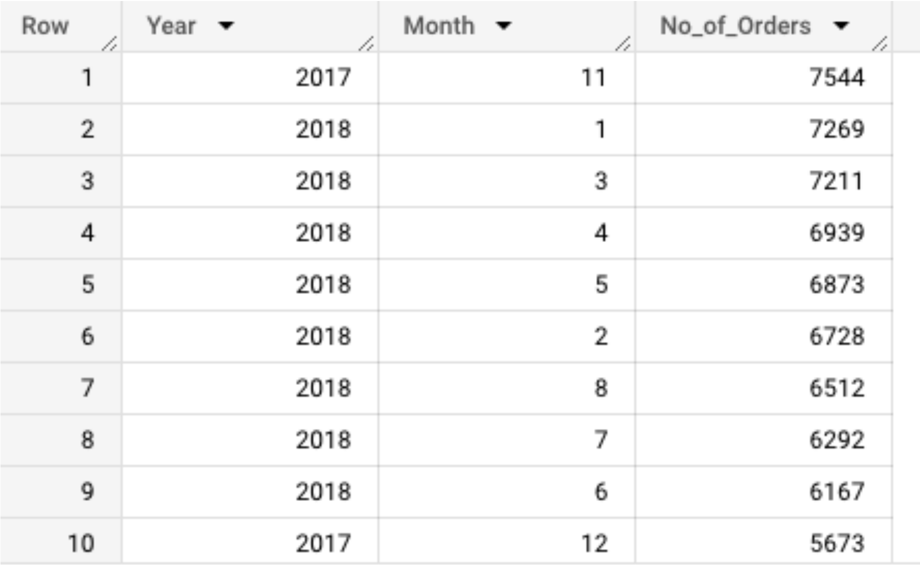
Insights :

From the above analysis, we can conclude that the sale of Target was maximum in the last quarter of the year i.e. between October to December and in the first quarter of the year i.e. January to March.

Answer (b):

```
select extract(year from order_purchase_timestamp) as Year,
extract (month from order_purchase_timestamp) as Month,
count(order_id) as No_of_Orders
from `target_store.orders`
group by Year, Month
order by No_of_Orders desc
```

Result :



Row	Year	Month	No_of_Orders
1	2017	11	7544
2	2018	1	7269
3	2018	3	7211
4	2018	4	6939
5	2018	5	6873
6	2018	2	6728
7	2018	8	6512
8	2018	7	6292
9	2018	6	6167
10	2017	12	5673

Insights :

Above result shows that in the Target received maximum orders in Nov’2017 followed by Jan’2018, March’2018, April’2018.

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

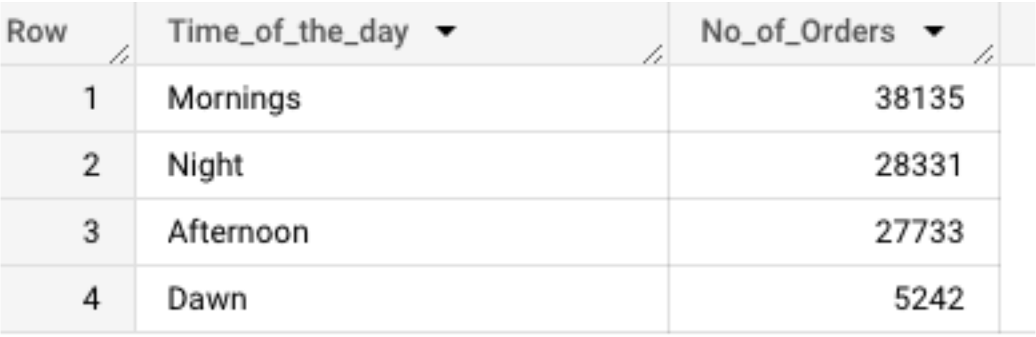
- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

Answer :

```
select
case
when extract(hour from order_purchase_timestamp) between 00 and 06
then 'Dawn'
when extract(hour from order_purchase_timestamp) between 07 and 12
then 'Afternoon'
```

```
when extract(hour from order_purchase_timestamp) between 13 and 18
then 'Mornings'
else 'Night'
end as Time_of_the_day, count(order_id) as No_of_Orders
from `target_store.orders`
group by Time_of_the_day
order by No_of_Orders desc
```

Result:



Row	Time_of_the_day	No_of_Orders
1	Mornings	38135
2	Night	28331
3	Afternoon	27733
4	Dawn	5242

Insights :

From the above analysis we can conclude that In Brazil, most of the orders placed in the morning time followed by night, then afternoon and then dawn. So, we can say least order get placed at midnight.

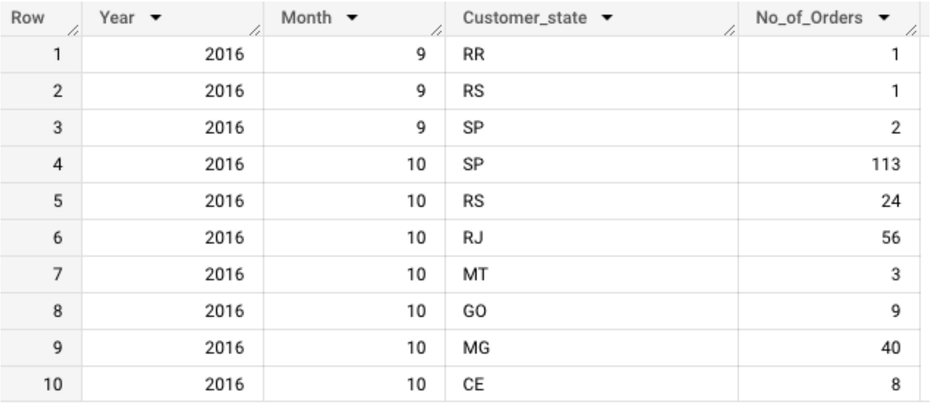
Question - 3 - Evolution of E-commerce orders in the Brazil region:

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

Answer :

```
select
extract(year from o.order_purchase_timestamp) as Year,
extract (month from o.order_purchase_timestamp) as Month,
c.customer_state as Customer_state,
count(o.order_id) as No_of_Orders
from `target_store.orders` o
join `target_store.customer` c
on o.customer_id = c.customer_id
group by Year,Month, Customer_state
order by Year, Month
```

Result:



Row	Year	Month	Customer_state	No_of_Orders
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	SP	113
5	2016	10	RS	24
6	2016	10	RJ	56
7	2016	10	MT	3
8	2016	10	GO	9
9	2016	10	MG	40
10	2016	10	CE	8

Insights :

Above result shows us how many orders got placed from each state as per the year and month.

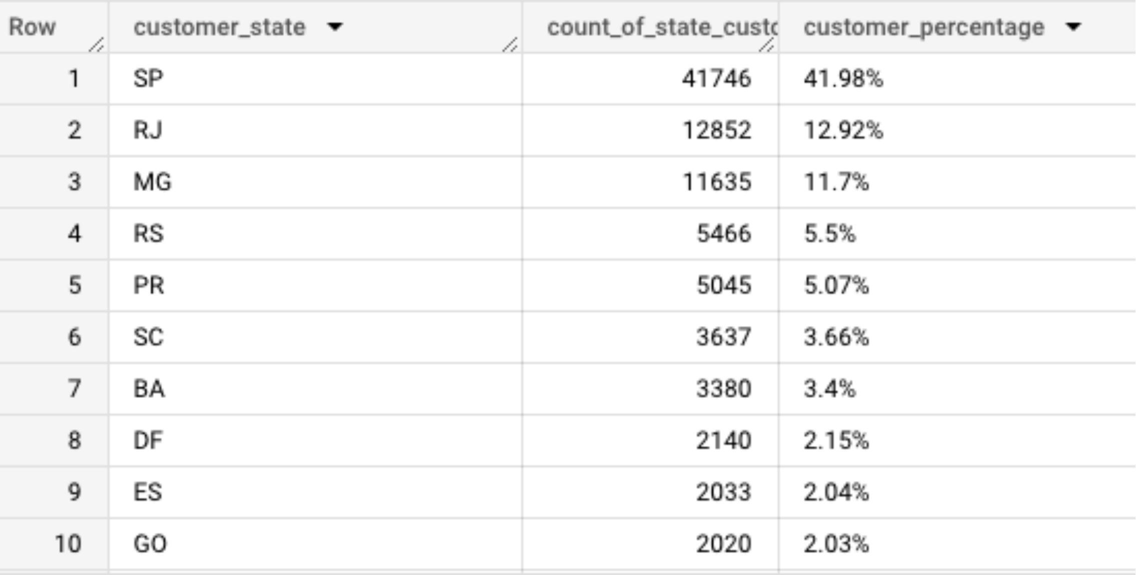
3.2 - How are the customers distributed across all the states?

Answer :

```
With total as
(select
  count(distinct customer_id) as total_customers
  from `target_store.customer`),
  state as
    (select
          customer_state, count(distinct c.customer_id) as
count_of_state_customers
     from `target_store.customer` c
    group by customer_state)

select customer_state, count_of_state_customers,
concat(round(100*(count_of_state_customers/
total_customers),2),'%') as customer_percentage
from total, state
order by count_of_state_customers desc
```

Result :



Row	customer_state	count_of_state_cust	customer_percentage
1	SP	41746	41.98%
2	RJ	12852	12.92%
3	MG	11635	11.7%
4	RS	5466	5.5%
5	PR	5045	5.07%
6	SC	3637	3.66%
7	BA	3380	3.4%
8	DF	2140	2.15%
9	ES	2033	2.04%
10	GO	2020	2.03%

Insights :

From above analysis, we can conclude that most of the customers are from Southeast Region and least customers are from North region of the Brazil.

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

4.1 - Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

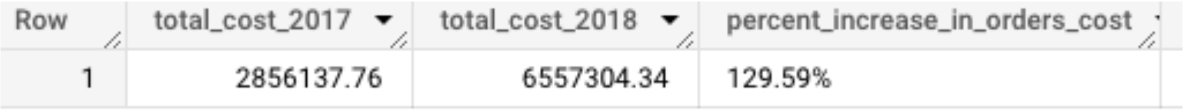
You can use the "payment_value" column in the payments table to get the cost of orders.

Answer :

```
   with cte_2017 as
  (select
    round(sum(p.payment_value),2) as total_cost_2017
  from `target_store.orders` o
  join `target_store.payments` p
  on o.order_id = p.order_id
  where  extract(year  from  o.order_purchase_timestamp)  =  2017  and
  extract(month from o.order_purchase_timestamp) in (2,3,4,5,6,7)),
  cte_2018 as
  (select
  round(sum(p.payment_value),2) as total_cost_2018
  from `target_store.orders` o
  join `target_store.payments` p
  on o.order_id = p.order_id
  where  extract(year  from  o.order_purchase_timestamp)  =  2018  and
  extract(month from o.order_purchase_timestamp) in (2,3,4,5,6,7))

  select *,
      concat(round(100*(total_cost_2018-total_cost_2017)/
  aa.total_cost_2017,2),'%') as percent_increase_in_orders_cost
  from
  (select total_cost_2017, total_cost_2018 from cte_2017, cte_2018 )aa
```

Result :



Row	total_cost_2017	total_cost_2018	percent_increase_in_orders_cost
1	2856137.76	6557304.34	129.59%

Insights :

From the above calculation, we can conclude that orders got increased approx. 130% from 2017 to 2018.

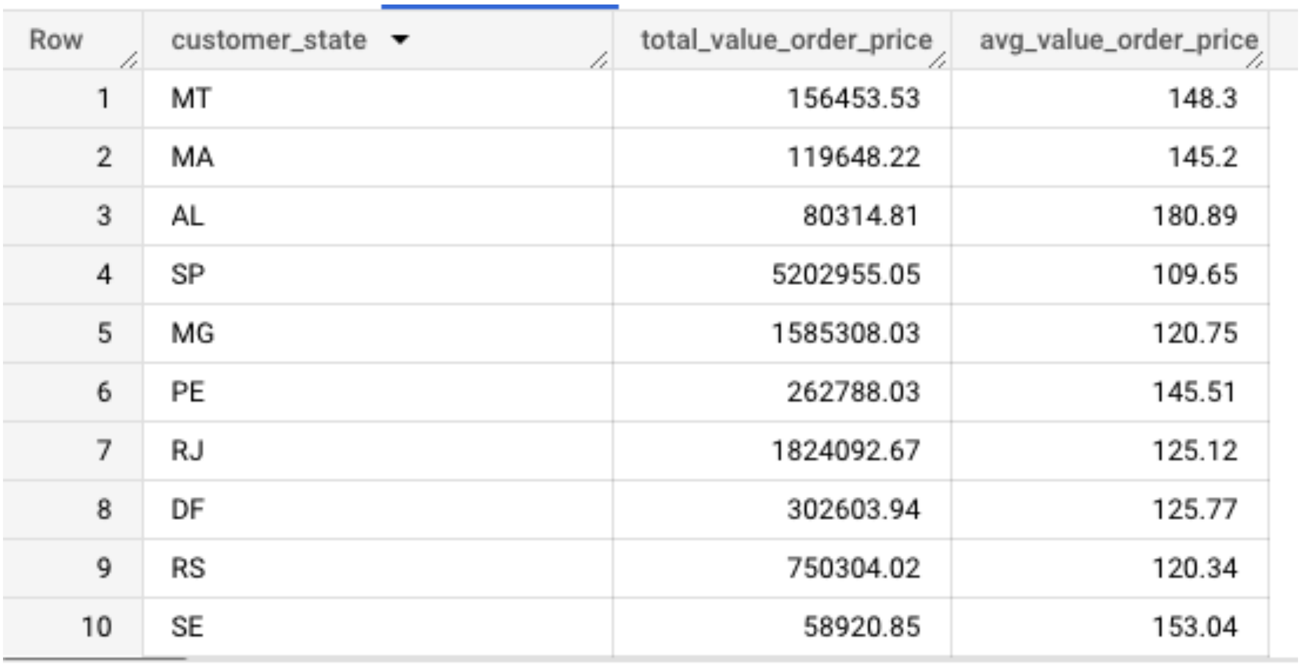
4.2 - Calculate the Total & Average value of order price for each state.

Answer :

```
    Select
    c.customer_state,   round(sum(oi.price),2)   as
    total_value_order_price,
      round(avg(oi.price),2) as avg_value_order_price
    from `target_store.order_items` oi
    join `target_store.orders` o
```

```
on oi.order_id = o.order_id
join `target_store.customer` c
on o.customer_id = c.customer_id
group by c.customer_state
```

Result :



Row	customer_state	total_value_order_price	avg_value_order_price
1	MT	156453.53	148.3
2	MA	119648.22	145.2
3	AL	80314.81	180.89
4	SP	5202955.05	109.65
5	MG	1585308.03	120.75
6	PE	262788.03	145.51
7	RJ	1824092.67	125.12
8	DF	302603.94	125.77
9	RS	750304.02	120.34
10	SE	58920.85	153.04

Insights :

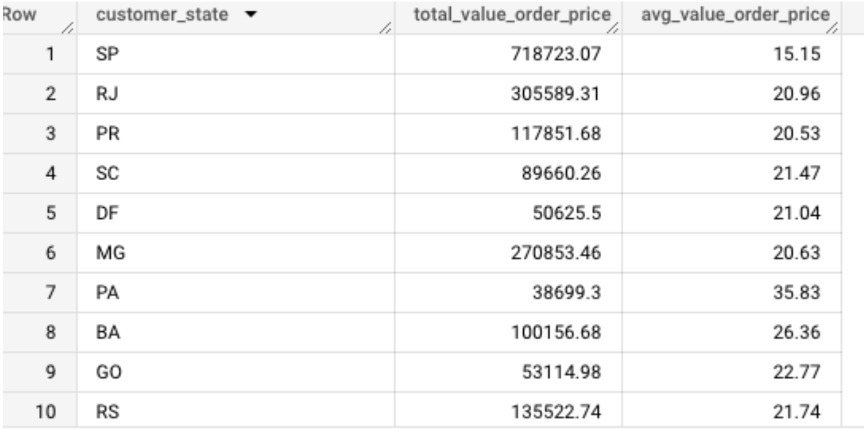
Above data represents total value of order and average value of orders against each state.

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

Answer:

```
select
c.customer_state, round(sum(oi.freight_value),2) as
total_value_order_price, round(avg(oi.freight_value),2) as
avg_value_order_price
from `target_store.order_items` oi
join `target_store.orders` o
on oi.order_id = o.order_id
join `target_store.customer` c
on o.customer_id = c.customer_id
group by c.customer_state
```

Result :



Row	customer_state	total_value_order_price	avg_value_order_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
7	PA	38699.3	35.83
8	BA	100156.68	26.36
9	GO	53114.98	22.77
10	RS	135522.74	21.74

Insights :

Above data represents total freight value of order and average freight value of orders against each state.

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

5.1 - 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.

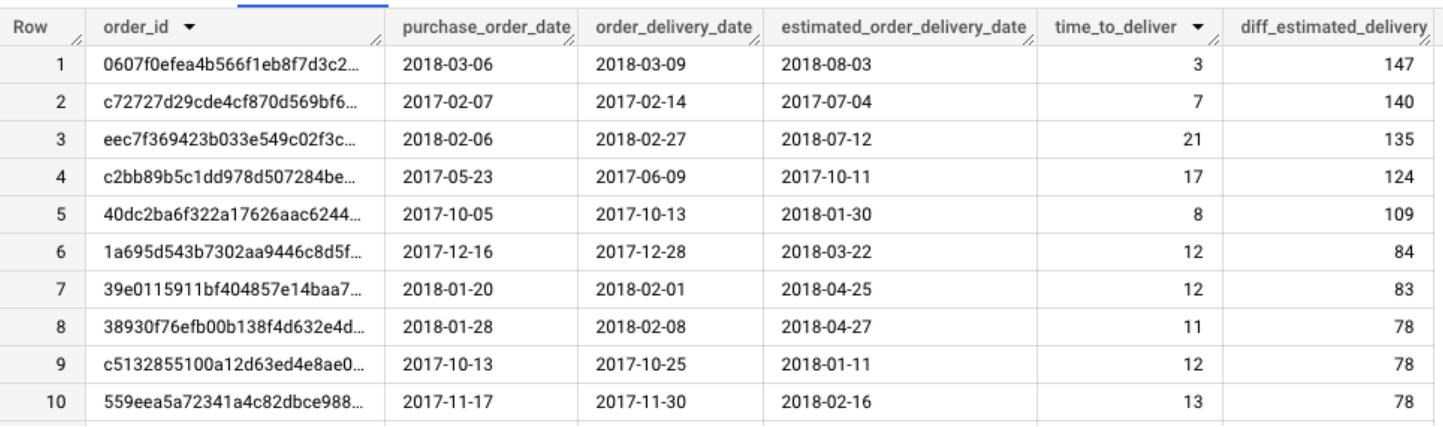
You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

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

Answer :

```
select
distinct order_id,
extract(date from order_purchase_timestamp) as
purchase_order_date,
extract(date from order_delivered_customer_date) as
order_delivery_date,
extract(date from order_estimated_delivery_date) as
estimated_order_delivery_date,
date_diff(extract(date from
order_delivered_customer_date),extract(date from
order_purchase_timestamp),day) as time_to_deliver,
date_diff(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),day) as diff_estimated_delivery
from `target_store.orders`
where extract(date from order_delivered_customer_date) is not null
order by diff_estimated_delivery desc
```

Result :



Row	order_id	purchase_order_date	order_delivery_date	estimated_order_delivery_date	time_to_deliver	diff_estimated_delivery
1	0607f0efea4b566f1eb8f7d3c2...	2018-03-06	2018-03-09	2018-08-03	3	147
2	c72727d29cde4cf870d569bf6...	2017-02-07	2017-02-14	2017-07-04	7	140
3	eec7f369423b033e549c02f3c...	2018-02-06	2018-02-27	2018-07-12	21	135
4	c2bb89b5c1dd978d507284be...	2017-05-23	2017-06-09	2017-10-11	17	124
5	40dc2ba6f322a17626aac6244...	2017-10-05	2017-10-13	2018-01-30	8	109
6	1a695d543b7302aa9446c8d5f...	2017-12-16	2017-12-28	2018-03-22	12	84
7	39e0115911bf404857e14baa7...	2018-01-20	2018-02-01	2018-04-25	12	83
8	38930f76efb00b138f4d632e4d...	2018-01-28	2018-02-08	2018-04-27	11	78
9	c5132855100a12d63ed4e8ae0...	2017-10-13	2017-10-25	2018-01-11	12	78
10	559eea5a72341a4c82dbce988...	2017-11-17	2017-11-30	2018-02-16	13	78

Insights :

Above result shows time taken to deliver orders against estimated time of delivery.

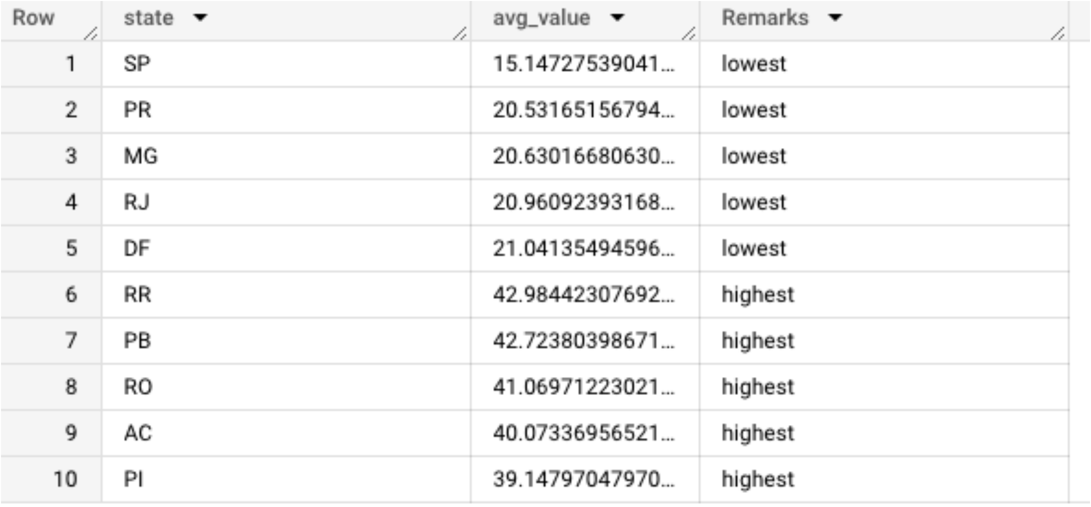
5.2 - Find out the top 5 states with the highest & lowest average freight value.

Answer (a):

```
with cte_top as
(select state, avg_value, 'highest' as Remarks from
  (select
state, avg_value,
row_number() over (order by avg_value desc) as rnk
from
(select c.customer_state as state,
avg(oi.freight_value) as avg_value
from `target_store.order_items` oi
join `target_store.orders` o
on oi.order_id = o.order_id
join `target_store.customer` c
on o.customer_id = c.customer_id
group by state
)bb
)aa
where rnk <= 5
order by avg_value),
cte_bottom as
(select
c.customer_state as state,
avg(oi.freight_value) as avg_freight, 'lowest' as Remarks
from `target_store.order_items` oi
join `target_store.orders` o
on oi.order_id = o.order_id
join `target_store.customer` c
on o.customer_id = c.customer_id
group by state
order by avg_freight asc
limit 5)

select * from cte_top
union distinct
select * from cte_bottom
```

Result:



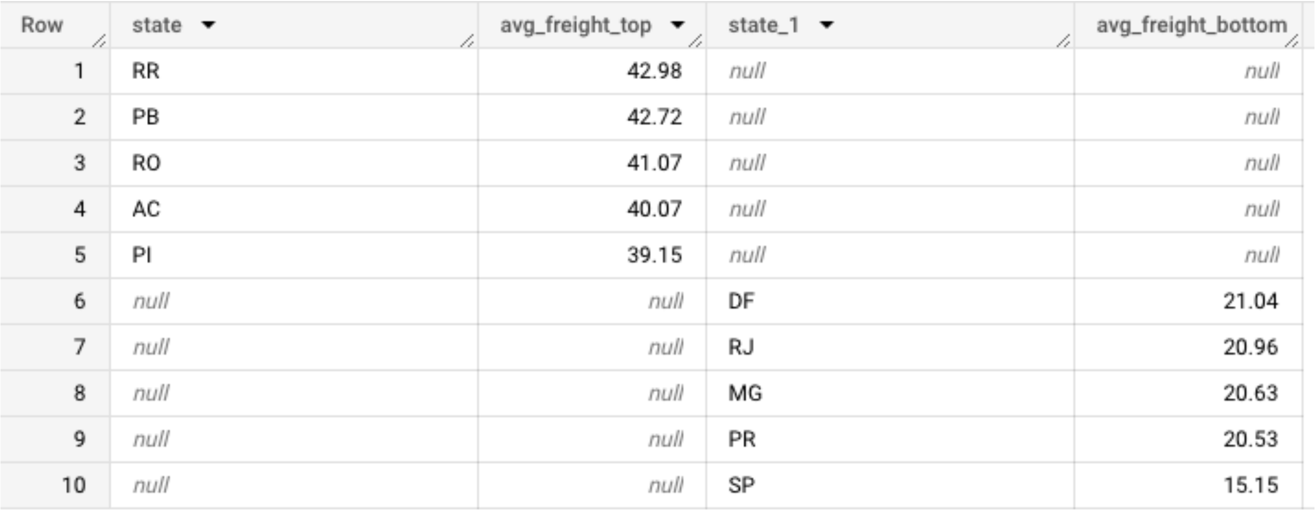
Row	state	avg_value	Remarks
1	SP	15.14727539041...	lowest
2	PR	20.53165156794...	lowest
3	MG	20.63016680630...	lowest
4	RJ	20.96092393168...	lowest
5	DF	21.04135494596...	lowest
6	RR	42.98442307692...	highest
7	PB	42.72380398671...	highest
8	RO	41.06971223021...	highest
9	AC	40.07336956521...	highest
10	PI	39.14797047970...	highest

Answer (b) :

```
with cte_top as
(select
c.customer_state as state,
round(avg(oi.freight_value),2) as avg_freight_top
from `target_store.order_items` oi
join `target_store.orders` o
on oi.order_id = o.order_id
join `target_store.customer` c
on o.customer_id = c.customer_id
group by state
order by avg_freight_top desc
limit 5),
cte_bottom as
(select
c.customer_state as state,
round(avg(oi.freight_value),2) as avg_freight_bottom
from `target_store.order_items` oi
join `target_store.orders` o
on oi.order_id = o.order_id
join `target_store.customer` c
on o.customer_id = c.customer_id
group by state
order by avg_freight_bottom asc
limit 5)
```

```
select  * from cte_top ct
full join cte_bottom cm
on ct.state = cm.state
order by avg_freight_top desc, avg_freight_bottom desc
```

Result :



Row	state	avg_freight_top	state_1	avg_freight_bottom
1	RR	42.98	*null*	*null*
2	PB	42.72	*null*	*null*
3	RO	41.07	*null*	*null*
4	AC	40.07	*null*	*null*
5	PI	39.15	*null*	*null*
6	*null*	*null*	DF	21.04
7	*null*	*null*	RJ	20.96
8	*null*	*null*	MG	20.63
9	*null*	*null*	PR	20.53
10	*null*	*null*	SP	15.15

Insights :

In above both the results we find top 5 highest & lowest average freight value state.

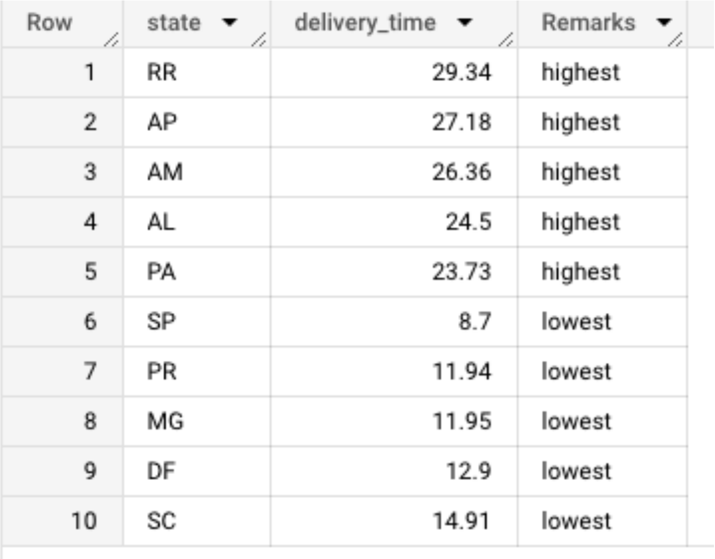
5.3 - Find out the top 5 states with the highest & lowest average delivery time.

Answer :

```
with cte_highest as
  (select c.customer_state as state,
          round(avg(date_diff(extract(date  from
o.order_delivered_customer_date),extract(date from
o.order_purchase_timestamp),day)),2) as delivery_time,
  'highest' as Remarks
from `target_store.orders` o
join `target_store.customer` c
on o.customer_id = c.customer_id
group by state
order by delivery_time desc
limit 5),
cte_lowest as
(select c.customer_state as state,
          round(avg(date_diff(extract(date  from
o.order_delivered_customer_date),extract(date from
o.order_purchase_timestamp),day)),2) as delivery_time,
  'lowest' as remarks
from `target_store.orders` o
join `target_store.customer` c
on o.customer_id = c.customer_id
group by state
order by delivery_time
limit 5)
```

```
select * from ate_highest
union distinct
select * from cte_lowest
```

Result :



Row	state	delivery_time	Remarks
1	RR	29.34	highest
2	AP	27.18	highest
3	AM	26.36	highest
4	AL	24.5	highest
5	PA	23.73	highest
6	SP	8.7	lowest
7	PR	11.94	lowest
8	MG	11.95	lowest
9	DF	12.9	lowest
10	SC	14.91	lowest

Insights :

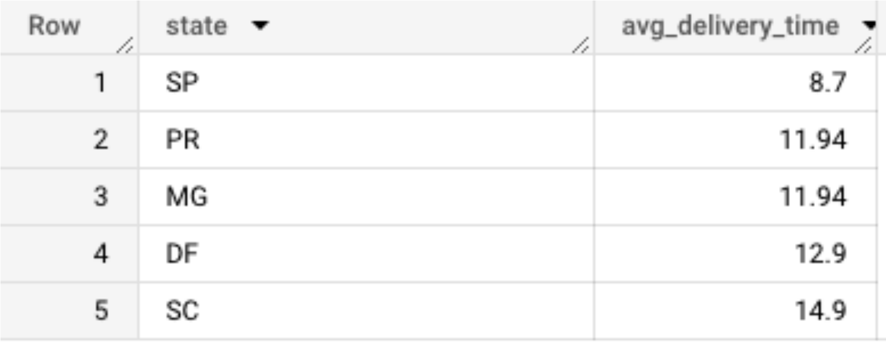
Above data shows top 5 states with highest and lowest average delivery time.

5.4 - 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.

Answer :

```
select
c.customer_state as state,
ROUND(avg(date_diff(extract(date from
o.order_delivered_customer_date),extract(date from
o.order_purchase_timestamp),day)),2) as avg_delivery_time
from `target_store.orders` o
join `target_store.customer` c
on o.customer_id = c.customer_id
where o.order_status = 'delivered'
group by state
order by avi_delivery_time
limit 5
```

Result :



Row	state	avg_delivery_time
1	SP	8.7
2	PR	11.94
3	MG	11.94
4	DF	12.9
5	SC	14.9

Insights :

Above are the top 5 states where delivery is comparatively very faster than other states against estimated delivery time.

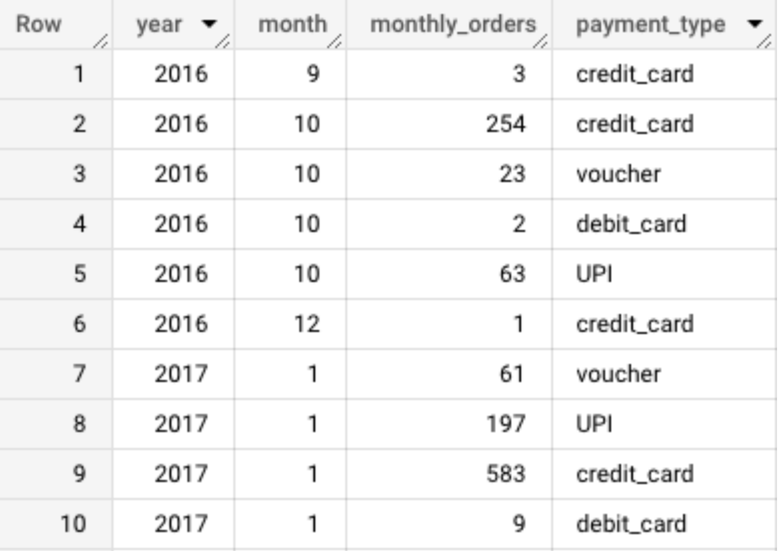
Question - 6 - Analysis based on the payments:

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

Answer :

```
select
extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
count(o.order_id) as monthly_orders,
p.payment_type as payment_type
from `target_store.payments` p
join `target_store.orders` o
on p.order_id = o.order_id
group by year, month, payment_type
order by year, month
```

Result :



Row	year	month	monthly_orders	payment_type
1	2016	9	3	credit_card
2	2016	10	254	credit_card
3	2016	10	23	voucher
4	2016	10	2	debit_card
5	2016	10	63	UPI
6	2016	12	1	credit_card
7	2017	1	61	voucher
8	2017	1	197	UPI
9	2017	1	583	credit_card
10	2017	1	9	debit_card

Insights :

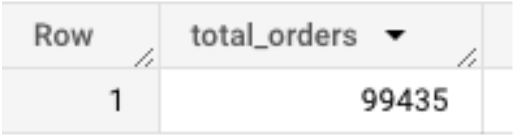
From above analysis we conclude that most of the orders got placed through credit cards.

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

Answer :

```
    select
count(distinct order_id) as total_orders
from `target_store.payments`
where payment_installments >= 1
```

Result :



Row	total_orders
1	99435

Insights :

Above result shows total orders on the basis of installment where amount is more that Rs. 0

Recommendations

After analysing this business case, my recommendation for the Company is below :

1. There is a scope of getting below details of customer to know customer in a better way -

a. customer_first_name,
b. customer_last_name,
c. customer_phone,
d. customer_sex,
e. customer_age,
f. customer_occupation

By getting all these details, we can segregate customer's group on the basis of sex, age-group, serviceman vs businessman and then can push relevant category for their preferred purchase by offering then some perks and discounts.

2. Most of the orders are getting placed in the last quarter of in the first quarter. So we need to manage our courier and transportation service according to ensure smooth operation.

3. Count of orders are less in the third quarter, so we need to offer some discounts, perks, cash back or any lucrative scheme to attract more customer.

4. As, most of the orders are getting places in the morning and night. So, we need to have very efficient transport service and need to put extra man power at that time to ensure smooth operations. In the dawn time, orders are least, so we can adjust manpower accordingly.

5. Customer base is least in North Brazil. We need to focus in this region.

6. Delivery time is good in Southeast Brazil and few areas of central east but for

rest regions, we need to focus more on courier and transportation service.

7. Most of the customers are using credit card for placing their orders. We can offer the discounts of card to maintain and even for more better customer base.

8. Least customers are selected debit card and voucher for payment. We need to focus for more on this to uplift customer base who can purchase through debit card and vouchers only.