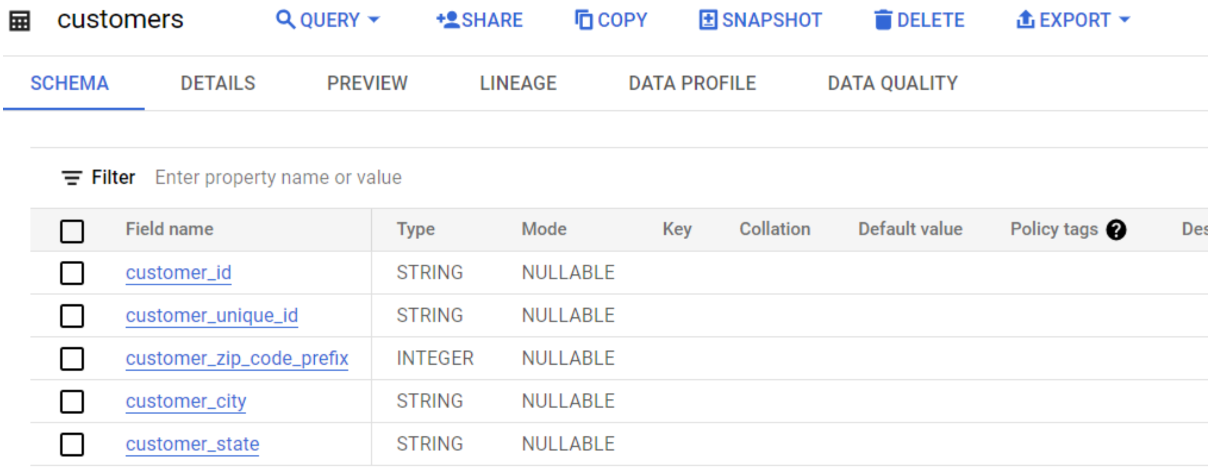
Note:- My databset Name is target_SQL
The dataset id is spry-starlight-393312.target_SQL

Q1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1. **The data type of all columns in the "customers" table.**
2. **Get the time range between which the orders were placed.**
3. **Count the Cities & States of customers who ordered during the given period.**

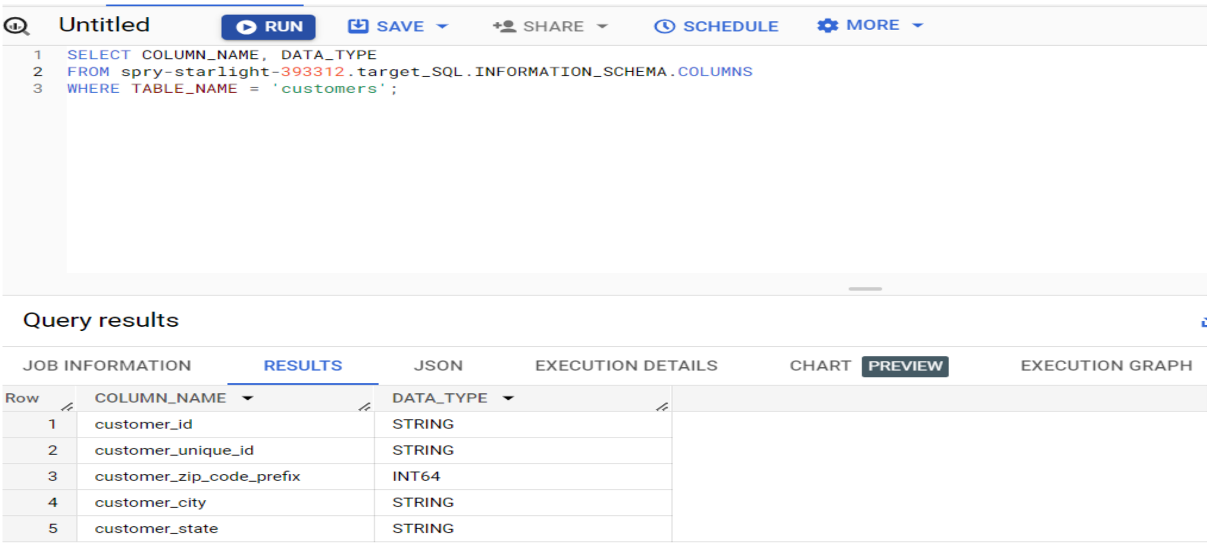
Ans. 1.1 Using Big query GUI



□	Field name	Type	Mode	Key	Collation	Default value	Policy tags ?	Des
□	customer_id	STRING	NULLABLE					
□	customer_unique_id	STRING	NULLABLE					
□	customer_zip_code_prefix	INTEGER	NULLABLE					
□	customer_city	STRING	NULLABLE					
□	customer_state	STRING	NULLABLE					

SQL QUERY:-

Using SQL query :- `SELECT COLUMN_NAME, DATA_TYPE FROM spry-starlight-393312.target_SQL.INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'customers';`



```
SELECT COLUMN_NAME, DATA_TYPE
FROM spry-starlight-393312.target_SQL.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers';
```

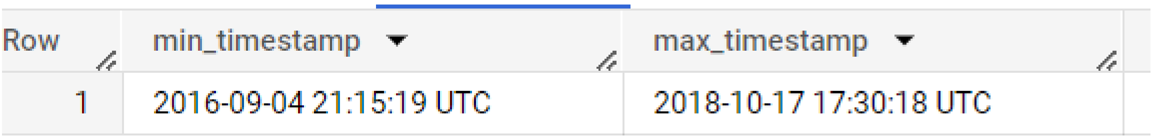
Row	COLUMN_NAME	DATA_TYPE
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

`

Inference :- understanding data types of table column helps in better understanding,interpretation and analysis of the data.

1.2 SQL query:-

```
select min(order_purchase_timestamp) as min_timestamp,
max(order_purchase_timestamp) as max_timestamp
from target_SQL.orders
```



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

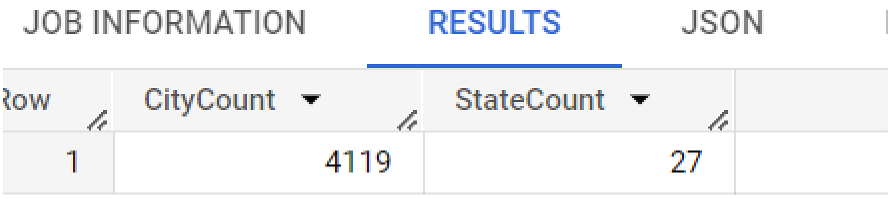
Inference:- The first purchase was done on 2016-09-04 21:15:19 UTC

And the last purchase was done on 2018-10-17 17:30:18 UTC

Acc to the given data. This time range helps us in understanding range of data in which orders were placed.

1.3 SQL Query :-

```
select count(distinct customer_city) as CityCount,
count(distinct customer_state) as StateCount
from target_SQL.customers c join
target_SQL.orders o
on c.customer_id=o.customer_id
```



Row	CityCount	StateCount
1	4119	27

Inference:- There are customers from 27 different states and 4119 different cities. We can understand extent of business expansion from this.

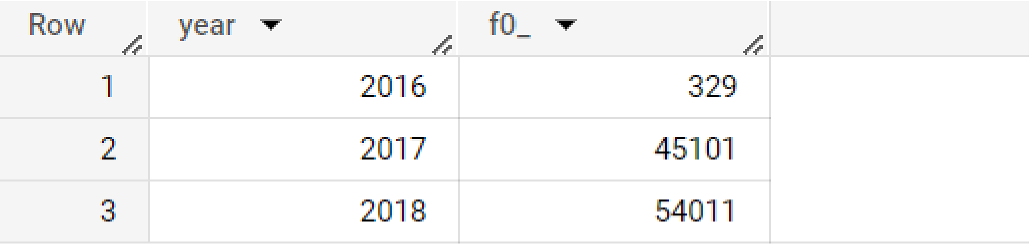
Q2.In-depth Exploration:

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

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

2.1 SQL Query :-

```
select extract(year from order_purchase_timestamp)as year, count(order_id)
from `target_SQL.orders`
group by (year) order by year;
```

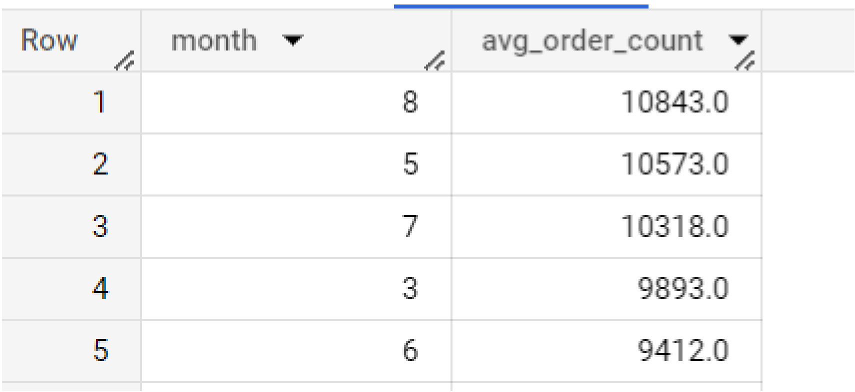


Row	year	f0_
1	2016	329
2	2017	45101
3	2018	54011

Inference:- Yes, there is a growing trend in no of order placed. From 2016 to 2017 it is exponential growth. From 2017-2018 the rate of growth has reduced but growth is still there.

2.2 SQL Query

```
select month,
avg(no_of_orders) as avg_order_count from
  (select extract(month from order_purchase_timestamp)
    as month,count(order_id) as no_of_orders
    from target_SQL.orders group by month )
group by month
order by avg_order_count desc
```



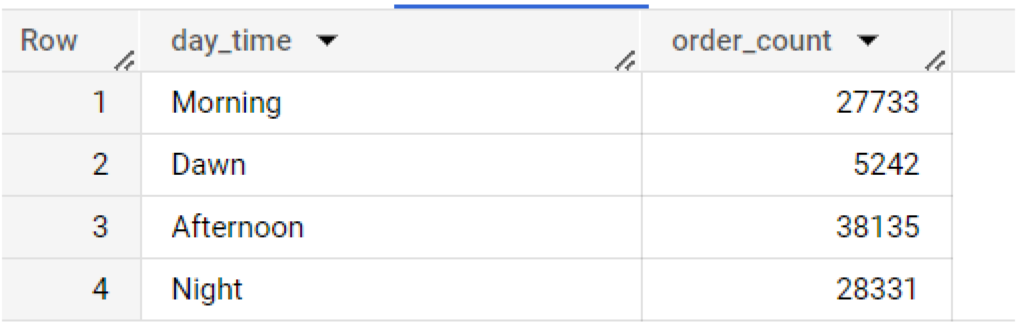
Row	month	avg_order_count
1	8	10843.0
2	5	10573.0
3	7	10318.0
4	3	9893.0
5	6	9412.0

Inference:- as we can see top sales happens in 8th month. Or further can say that 5,6,7,8th month is the peak season. 9,10,11,12th month is the lowest sale season.

2.3 SQL Query

```
with CTE as (
  select extract(hour from order_purchase_timestamp) as day_hour,
  order_id
  from target_SQL.orders
)

select
  case when day_hour between 0 and 6 then "Dawn"
  when day_hour between 7 and 12 then "Morning"
  when day_hour between 13 and 18 then "Afternoon"
  when day_hour between 19 and 23 then "Night"
  End as day_time,
  count( distinct order_id) as order_count,
from CTE
group by day_time
```



Row	day_time	order_count
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

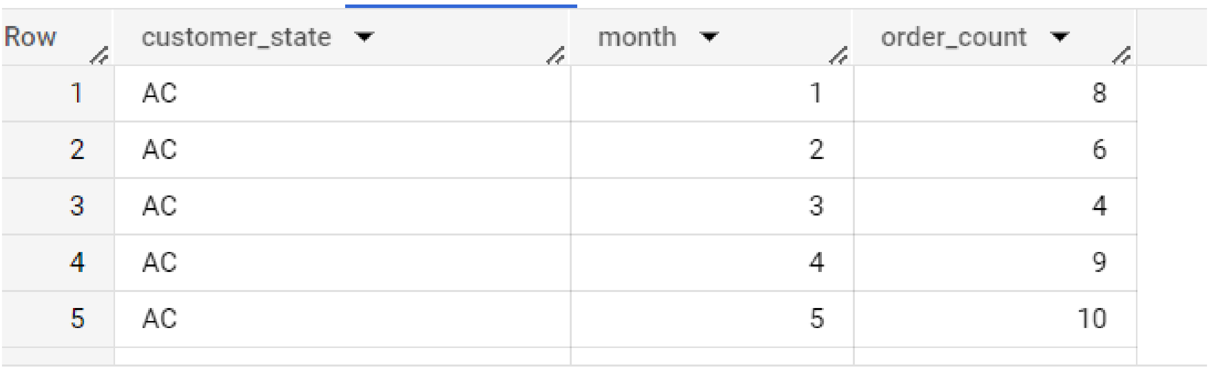
Inference: The highest orders were placed during the afternoon, whereas the lowest were placed during the Dawn . this means brazilian customers like to place order during their leisure time in afternoon.

Q3.Evolution of E-commerce orders in the Brazil region:

1. **Get the month-on-month no. of orders placed in each state.**
2. **How are the customers distributed across all the states?**

3.1 SQL Query

```
select c.customer_state, extract(month from o.order_purchase_timestamp) as
month,count(distinct order_id) as order_count
from target_SQL.customers c
join target_SQL.orders o
on c.customer_id=o.customer_id
group by c.customer_state,month
order by order_count desc
```

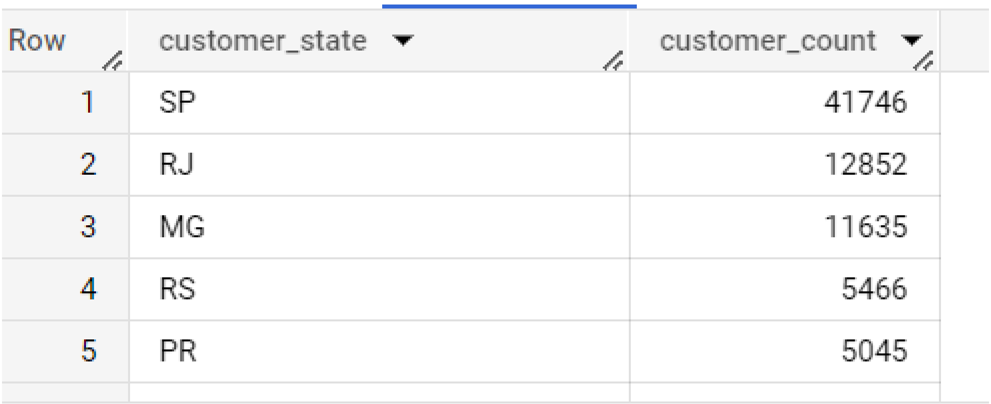


Row	customer_state	month	order_count
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10

Inference: Every month, the highest no of orders were placed in the SP state of Brazil,followed by RJ and MG

3.2
SQL Query :-

```
select customer_state, count(customer_id) as customer_count
from target_SQL.customers
group by customer_state
order by customer_count desc
```



Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045

Inference:- highest no of customers belong to SP state followed by RJ and MG. While the lowest no of customers belongs to RR. This is due to the fact that SP is most populus state in Brazil. This indicates a positive correlation bw a state's population and its order count.

Q4. Impact on the Economy: Analyze the money movement by e-commerce by looking at order prices, freight, and others.

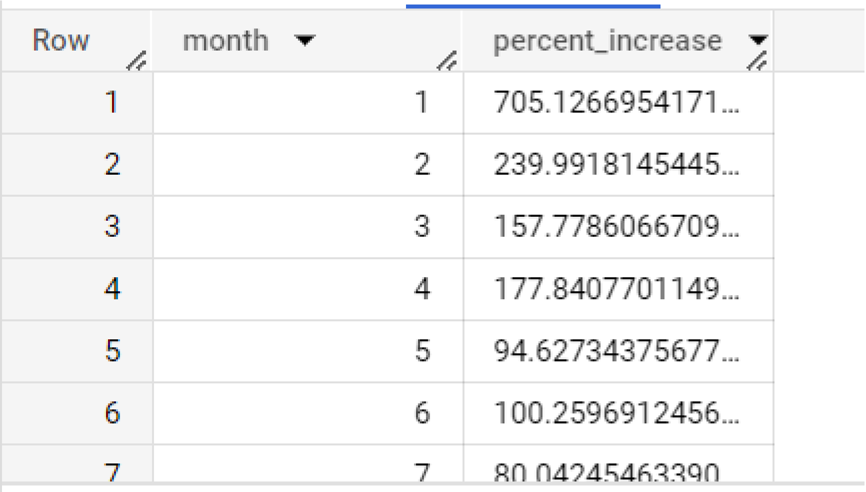
1. **Get the % increase in the cost of orders from the year 2017 to 2018 (include months between Jan to Aug only).**
   You can use the "payment_value" column in the payments table to get the cost of orders.
2. **Calculate the Total and average value of order price for each state.**
3. **Calculate the Total and average value of order freight for each state.**

4.1

SQL Query(month by month)

```
WITH CTE AS (
  SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
    SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017
             AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND
8
             THEN p.payment_value END) AS month_sale2017,
    SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2018
             AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND
8
             THEN p.payment_value END) AS month_sale2018
  FROM
    target_SQL.orders o
  JOIN
    target_SQL.payments p
  ON
    o.order_id = p.order_id
  WHERE
    EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8
  GROUP BY
    month
  order by month
)

SELECT
  month,
  ((month_sale2018 - month_sale2017) / month_sale2017) * 100 AS
percent_increase
FROM
  CTE;
```

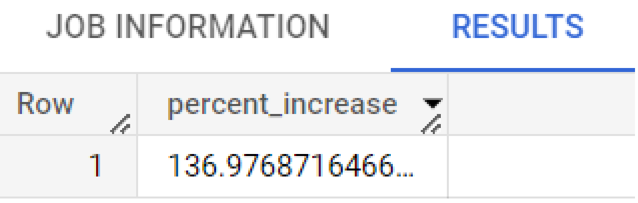


Row	month	percent_increase
1	1	705.1266954171...
2	2	239.9918145445...
3	3	157.7786066709...
4	4	177.8407701149...
5	5	94.62734375677...
6	6	100.2596912456...
7	7	80.04245463390

SQL Query(overall growth)

```
with CTE as (
  select sum(case when extract(year from order_purchase_timestamp)=2017
then p.payment_value else 0 end) as year2017,
  sum(case when extract(year from order_purchase_timestamp)=2018 then
p.payment_value else 0 end) as year2018
  from target_SQL.orders o join target_SQL.payments p
  on o.order_id=p.order_id
  where extract(month from order_purchase_timestamp) between 1 and 8
 )

 select (year2018-year2017)/year2017*100 as percent_increase
 from CTE
```

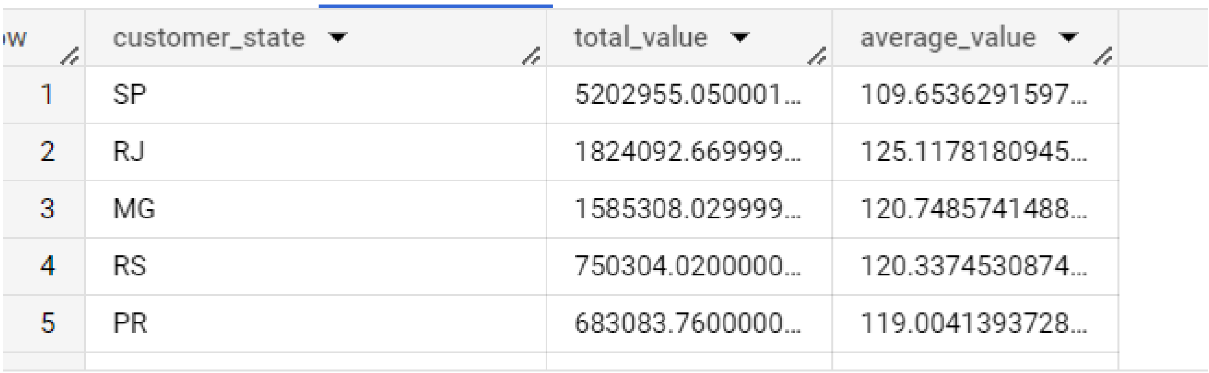


Row	percent_increase
1	136.9768716466...

Inference:- January shows the highest percentage increase, followed by February and April**,**There is almost a 137% increase in the cost of orders from year 2017 to 2018 (Jan to Aug only).

4.2
SQl Query

```
select c.customer_state,
sum(oi.price) as total_value,
avg(oi.price) as average_value
from target_SQL.customers c
join target_SQL.orders o
on c.customer_id=o.customer_id
join target_SQL.order_items oi
on oi.order_id=o.order_id
group by c.customer_state
order by total_value desc
```

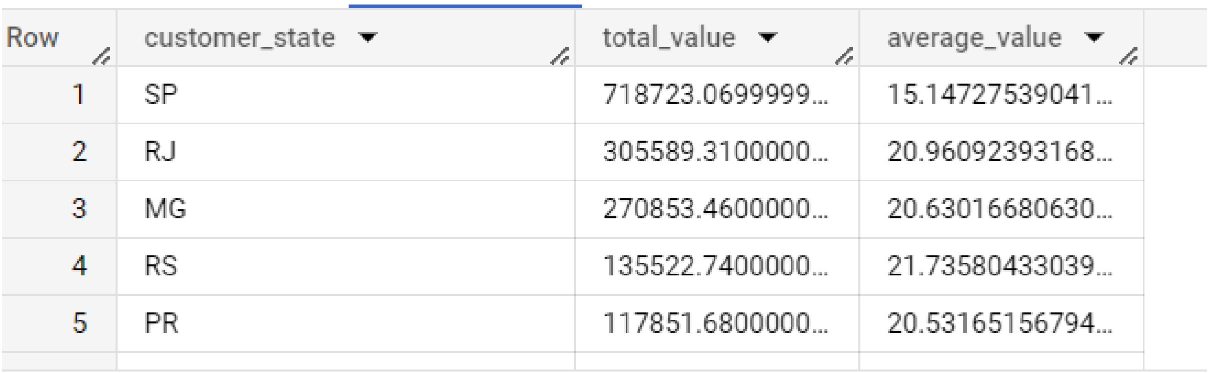


w	customer_state	total_value	average_value
1	SP	5202955.050001...	109.6536291597...
2	RJ	1824092.669999...	125.1178180945...
3	MG	1585308.029999...	120.7485741488...
4	RS	750304.0200000...	120.3374530874...
5	PR	683083.7600000...	119.0041393728...

Inference: The highest total value of orders was placed from the SP state, and lowest from RR. The highest average value of orders was placed from the PB state, and lowest from SP

4.3
SQL Query

```
select c.customer_state,
sum(oi.freight_value) as total_value,
sum(oi.freight_value)/count(*) as average_value
from target_SQL.customers c
join target_SQL.orders o
on c.customer_id=o.customer_id
join target_SQL.order_items oi
on oi.order_id=o.order_id
group by c.customer_state
order by total_value desc
```



Row	customer_state	total_value	average_value
1	SP	718723.0699999...	15.14727539041...
2	RJ	305589.3100000...	20.96092393168...
3	MG	270853.4600000...	20.63016680630...
4	RS	135522.7400000...	21.73580433039...
5	PR	117851.6800000...	20.53165156794...

Inference: The highest total freight value is in the SP state, while the lowest is in the RR.
The highest average freight value is in RR, while lowest average freight value is in SP.

Q5. Analysis based on sales, freight and delivery time.

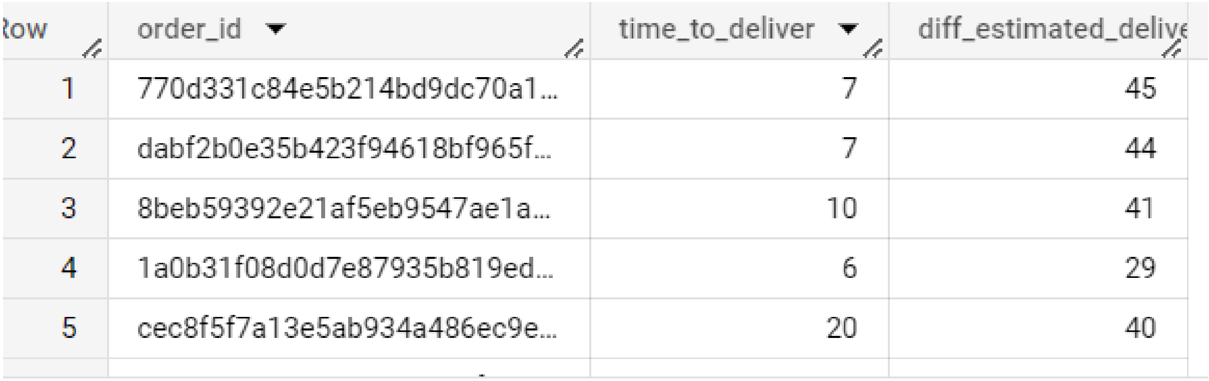
1. **Find the no. of days taken to deliver each order from the order's purchase date as delivery time.**
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
   Do this in a single query.

   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**
2. **Find out the top 5 states with the highest & lowest average freight value.**
3. **Find out the top 5 states with the highest & lowest average delivery time.**
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.

5.1

SQL Query

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

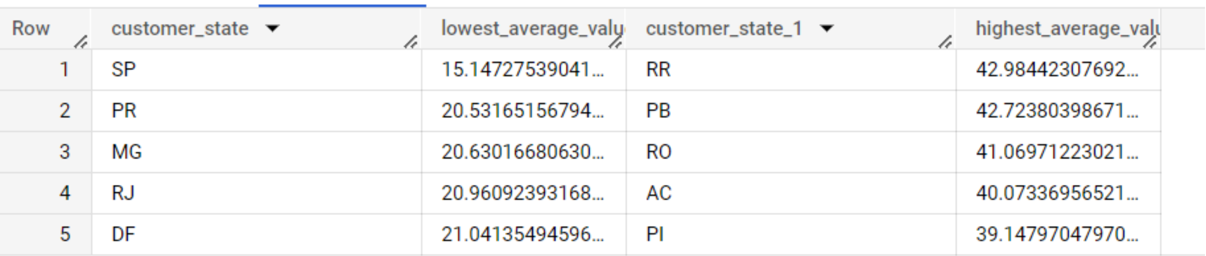


Row	order_id	time_to_deliver	diff_estimated_delive
1	770d331c84e5b214bd9dc70a1...	7	45
2	dabf2b0e35b423f94618bf965f...	7	44
3	8beb59392e21af5eb9547ae1a...	10	41
4	1a0b31f08d0d7e87935b819ed...	6	29
5	cec8f5f7a13e5ab934a486ec9e...	20	40

5.2

SQL Query

```
WITH CTE AS (
  SELECT c.customer_state,AVG(oi.freight_value) AS average_value,
  ROW_NUMBER() OVER (ORDER BY AVG(oi.freight_value)) AS asc_rnk,
  ROW_NUMBER() OVER (ORDER BY AVG(oi.freight_value) DESC) AS desc_rnk
  FROM target_SQL.customers c
  JOIN target_SQL.orders o
  ON c.customer_id = o.customer_id
  JOIN target_SQL.order_items oi
  ON oi.order_id = o.order_id
  GROUP BY c.customer_state
)
SELECT
  T1.customer_state,T1.average_value AS lowest_average_value,
  T2.customer_state,T2.average_value AS highest_average_value
FROM (
  SELECT customer_state,average_value,asc_rnk
  FROM CTE
  ORDER BY average_value LIMIT 5
) AS T1
FULL JOIN (
  SELECT customer_state, average_value,desc_rnk
  FROM CTE
  ORDER BY average_value DESC LIMIT 5
) AS T2
ON T1.asc_rnk = T2.desc_rnk
ORDER BY COALESCE(T1.asc_rnk, T2.desc_rnk);
```



Row	customer_state	lowest_average_valu	customer_state_1	highest_average_valu
1	SP	15.14727539041…	RR	42.98442307692…
2	PR	20.53165156794…	PB	42.72380398671…
3	MG	20.63016680630…	RO	41.06971223021…
4	RJ	20.96092393168…	AC	40.07336956521…
5	DF	21.04135494596…	PI	39.14797047970…

Inference:- Top 5states with highest average freight values are SP,PR,MG,RJ,DF Whereas top5 states with lowest average freight values are RR, PB,RO, AC, PI

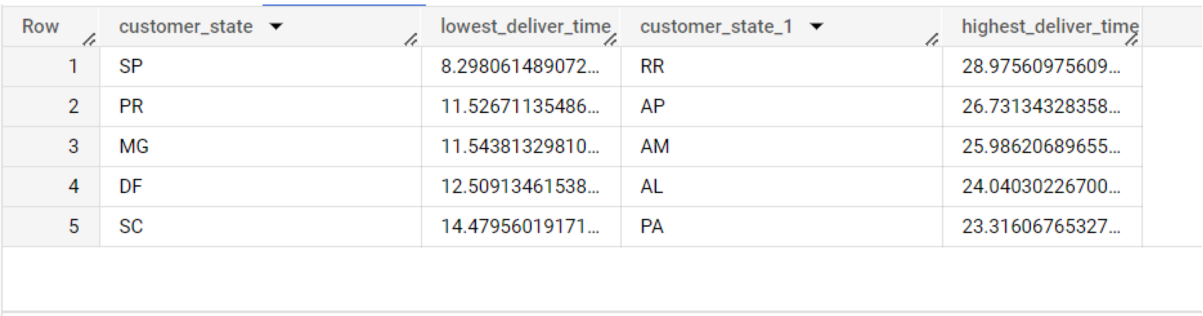
5.3

SQL Query:-

```
WITH CTE AS (
  SELECT c.customer_state,

AVG(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))
AS average_deliver_time,
  FROM target_SQL.customers c
  JOIN target_SQL.orders o
  ON c.customer_id = o.customer_id
  GROUP BY c.customer_state
)
SELECT
  T1.customer_state,T1.average_deliver_time AS lowest_deliver_time,
  T2.customer_state,T2.average_deliver_time AS highest_deliver_time
FROM (
  SELECT customer_state,average_deliver_time,
  row_number() over(order by average_deliver_time) as asc_rnk
  FROM CTE
  LIMIT 5
) AS T1
FULL JOIN (
  SELECT customer_state, average_deliver_time,
  row_number() over(order by average_deliver_time desc) as desc_rnk
  FROM CTE
  LIMIT 5
) AS T2
```

```
ON T1.asc_rnk = T2.desc_rnk
ORDER BY COALESCE(T1.asc_rnk, T2.desc_rnk);
```



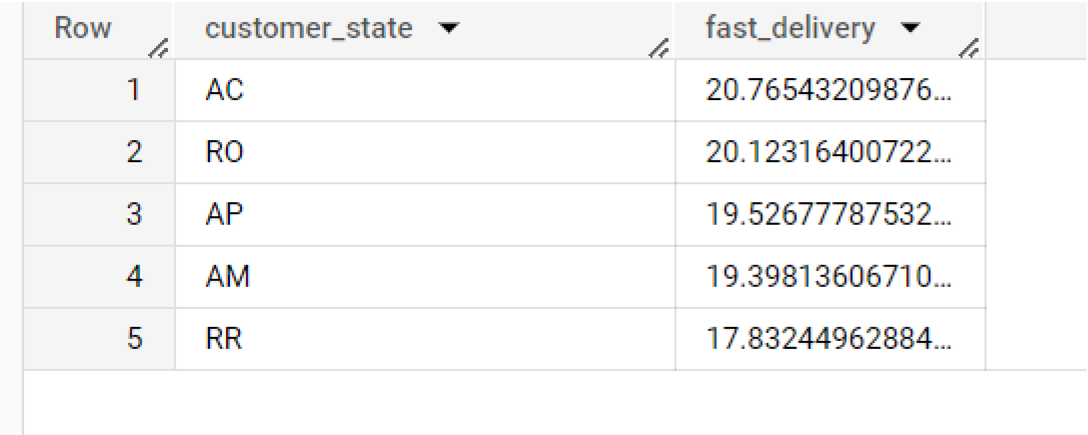
Row	customer_state	lowest_deliver_time	customer_state_1	highest_deliver_time
1	SP	8.298061489072…	RR	28.97560975609…
2	PR	11.52671135486…	AP	26.73134328358…
3	MG	11.54381329810…	AM	25.98620689655…
4	DF	12.50913461538…	AL	24.04030226700…
5	SC	14.47956019171…	PA	23.31606765327…

Inference:- Top 5states with highest average freight values are SP,PR,MG,DF,SC Whereas top5 states with lowest average freight values are RR, AP, AM, AL, PA

5.4

```
with CTE as(
  select
c.customer_state,avg(abs(date_diff(date(o.order_delivered_customer_date),da
te(o.order_purchase_timestamp),day))) as  avg_delivered_date,

avg(abs(date_diff(date(o.order_estimated_delivery_date),date(o.order_purcha
se_timestamp),day))) as avg_estimated_date
  from target_SQL.orders o
  join target_SQL.customers c
  on o.customer_id=c.customer_id
  group by c.customer_state
)
select customer_state,avg_estimated_date-avg_delivered_date as
fast_delivery
from CTE order by fast_delivery desc limit 5;
```



Row	customer_state	fast_delivery
1	AC	20.76543209876...
2	RO	20.12316400722...
3	AP	19.52677787532...
4	AM	19.39813606710...
5	RR	17.83244962884...

Inference :- top 5 states where the order delivery is really fast as compared to the estimated date of delivery AC, RO, AP, AM, RR

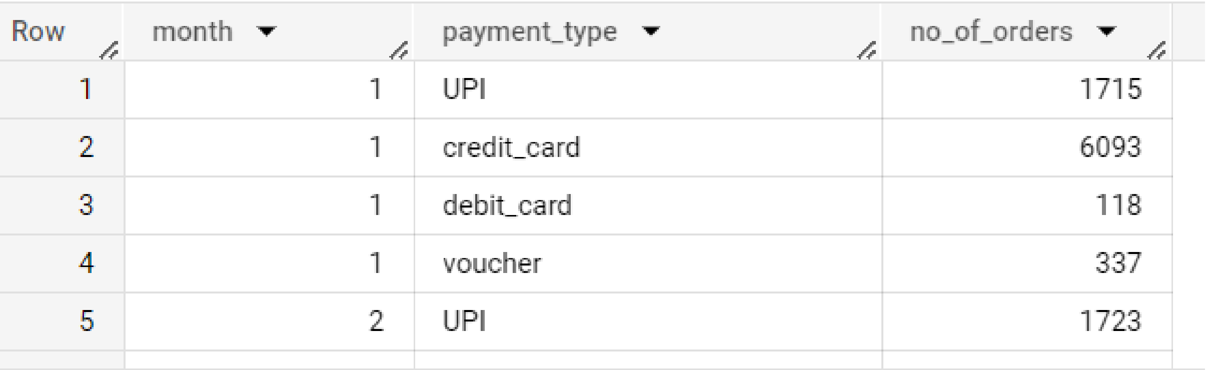
Q6. Analysis based on the payments:

1. **Find the month on month no. of orders placed using different payment types.**
2. **Find the no. of orders placed on the basis of the payment installments that have been paid.**

6.1

SQL Query

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



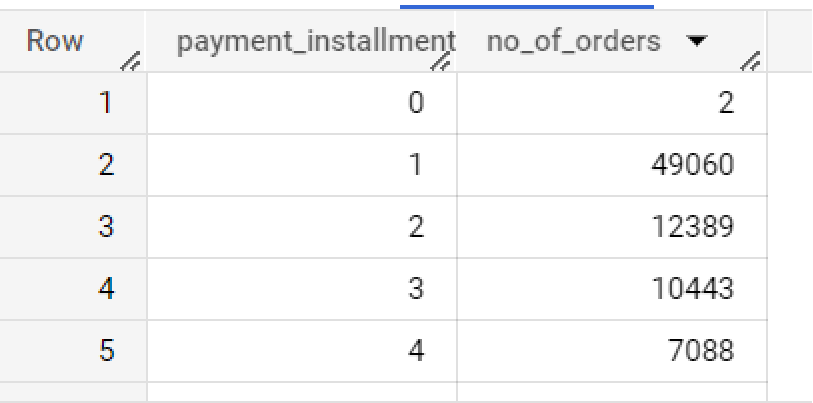
Row	month	payment_type	no_of_orders
1	1	UPI	1715
2	1	credit_card	6093
3	1	debit_card	118
4	1	voucher	337
5	2	UPI	1723

Inference:- Credit card is the used mode of transaction followed by UPI.

6.2

SQL Query

```
select p.payment_installments, count(distinct o.order_id) as no_of_orders
from target_SQL.orders o join target_SQL.payments p
on o.order_id=p.order_id
group by p.payment_installments
order by p.payment_installments,no_of_orders
```



Row	payment_installment	no_of_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088

Inference :- Most of the orders have only one installement paid. Max installements of any order is 24

INSIGHTS:-

- SP state is leading by great difference from other states in terms of number of orders. This indicates a need to improve business in other states.
- There is seasonality trend in no of orders places, business should keep this mind and can strategize marketing and sales according to the peak seasonality.
- Delivery time can be improved in some regions, this will lead to positive customer feedback and cusomter retention.
- Customer demographics plays an important role in planning business expansion and building marketing strategies.
- Off seasonality sales can be improved by deploying suitable discount schemes.(sep to Dec)

RECOMMENDATIONS:-

- Deliveries can be made faster by optimizing warehouse operations.

- Customer should be retened in top states(with high sales) using referrals and membership programs.
- Continuos reiteration of freight price to optimize it.
- Social media can be used as a tool to advertise products, promotion and building brand value.
- Customer support should be efficient to answer customer queries efficiently.