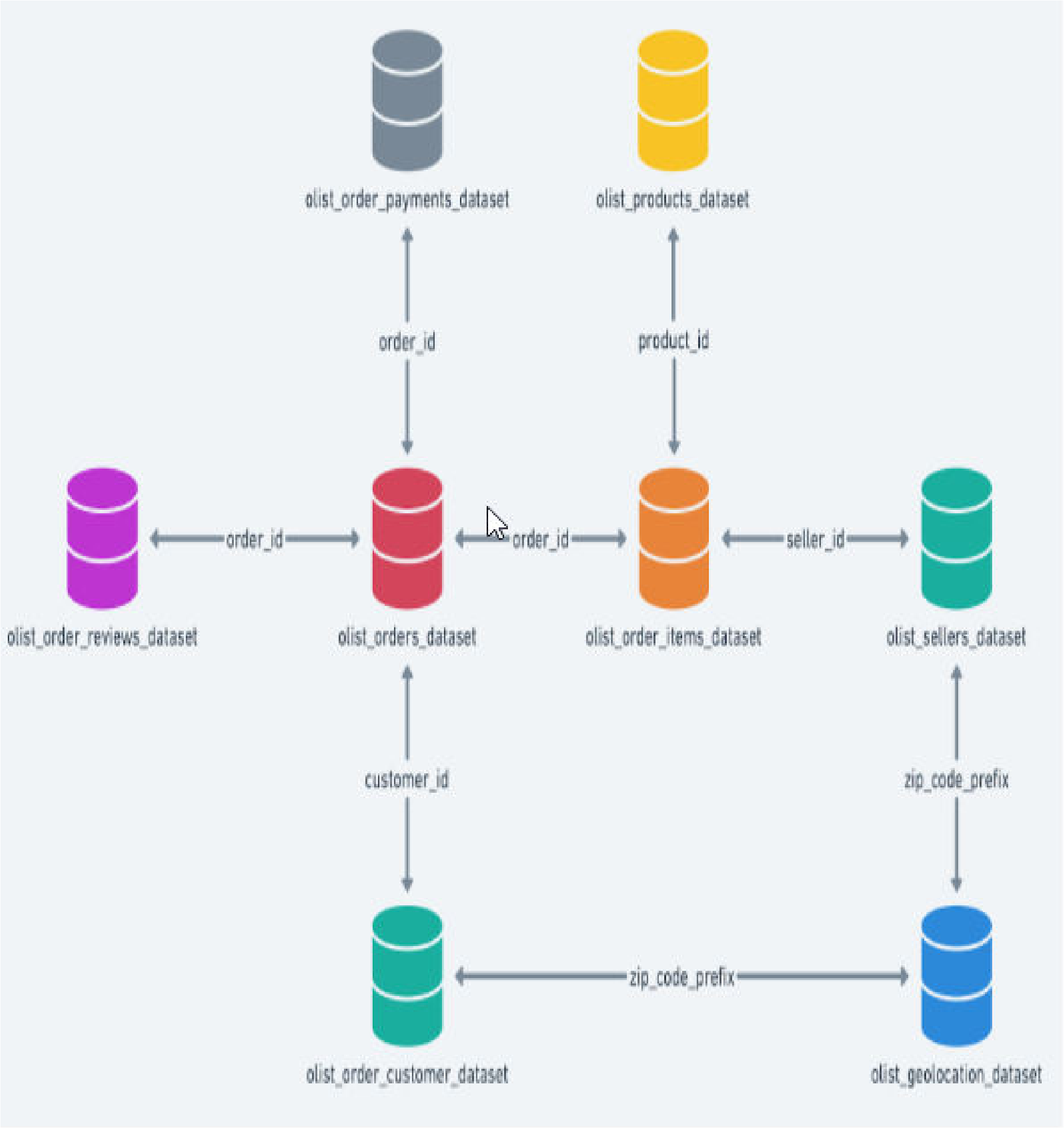


The data is available in 8 csv files:

1.customers.csv

2.sellers.csv

3.order_items.csv

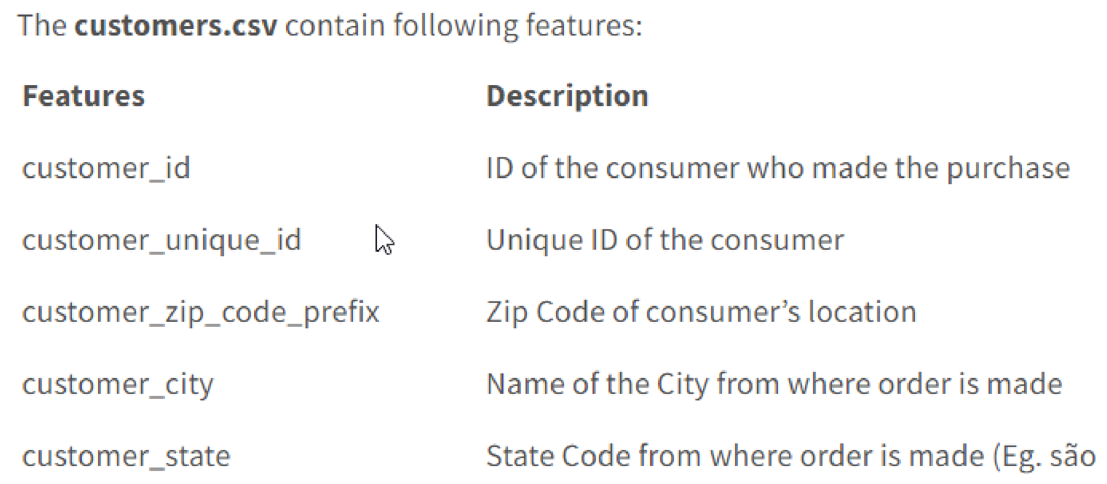
4.geolocation.csv

5.payments.csv

6.reviews.csv

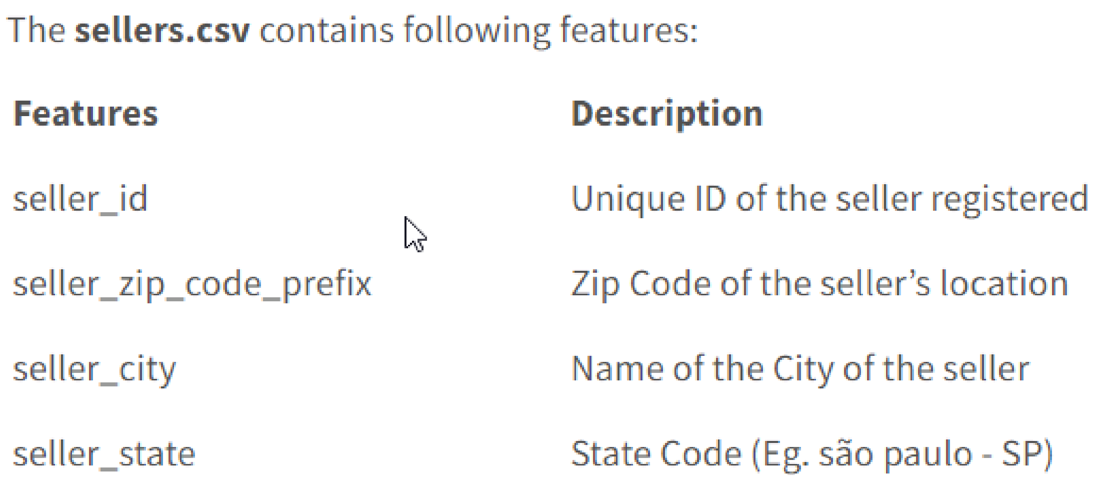
7.orders.csv

8.products.csv



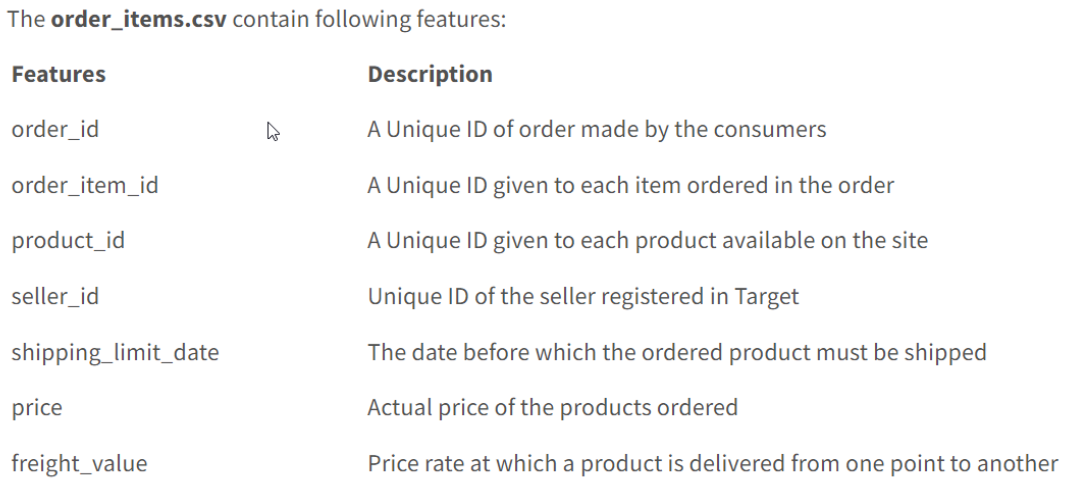
The **customers.csv** contain following features:

Features	Description
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer’s location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são



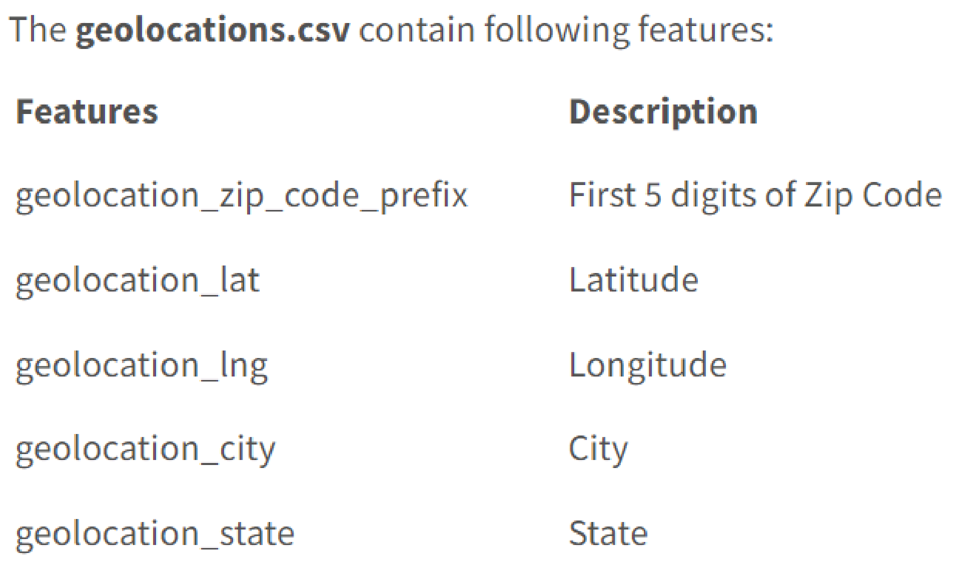
The **sellers.csv** contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller’s location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)



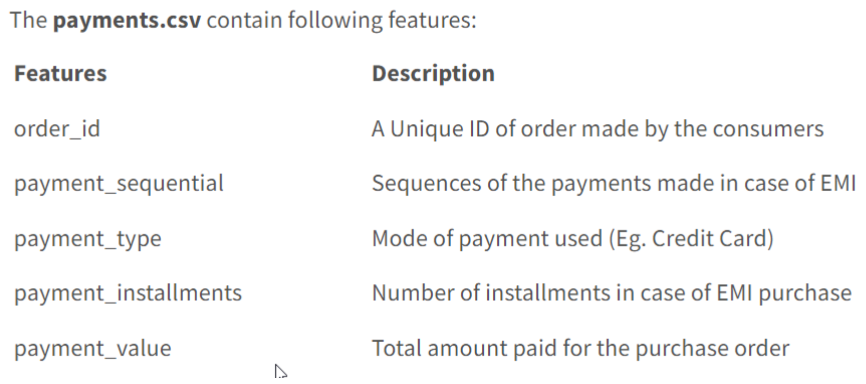
The **order_items.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another



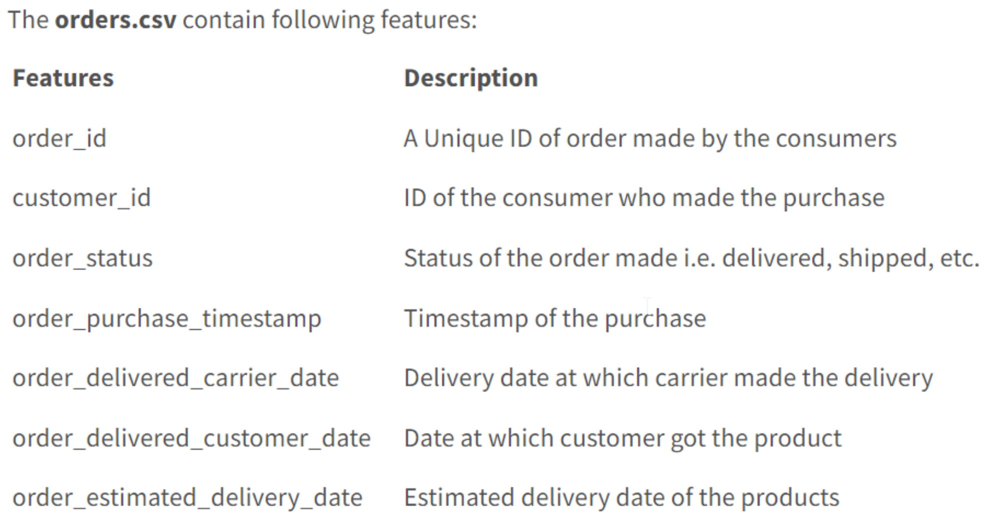
The **geolocations.csv** contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code
geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State



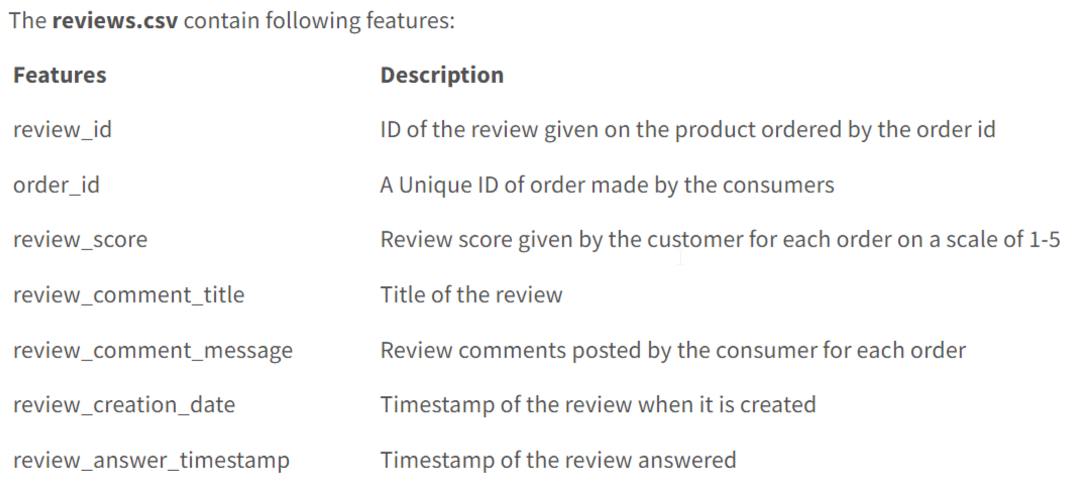
The **payments.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order



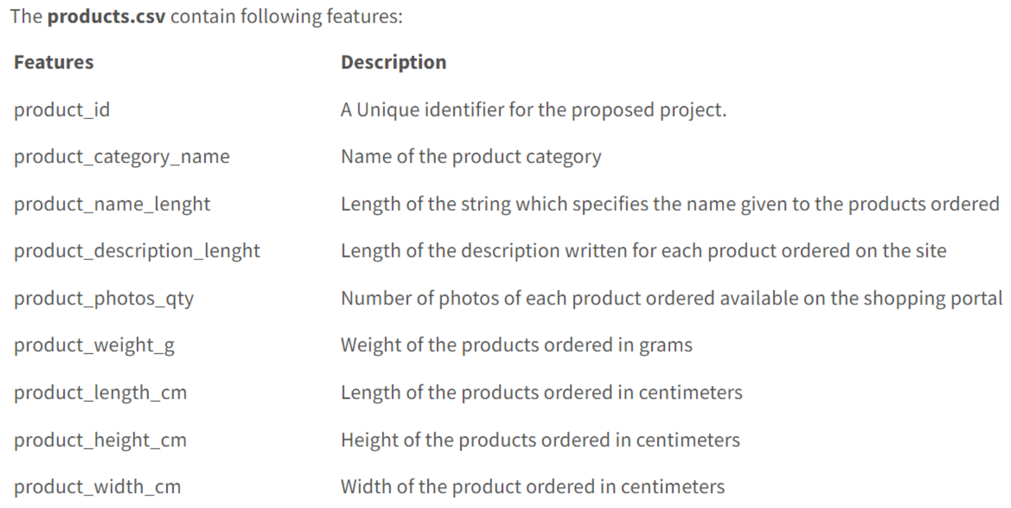
The **orders.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.
order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery
order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products



The **reviews.csv** contain following features:

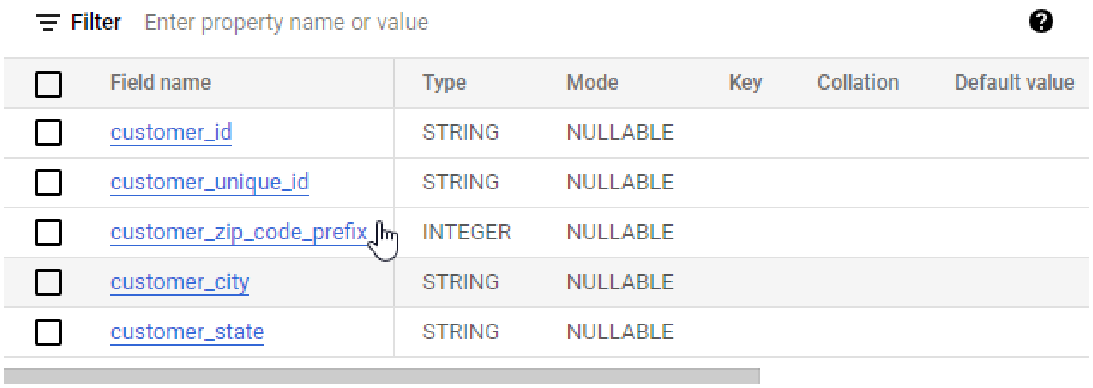
Features	Description
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered



The **products.csv** contain following features:

Features	Description
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	Length of the string which specifies the name given to the products ordered
product_description_lenght	Length of the description written for each product ordered on the site
product_photos_qty	Number of photos of each product ordered available on the shopping portal
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

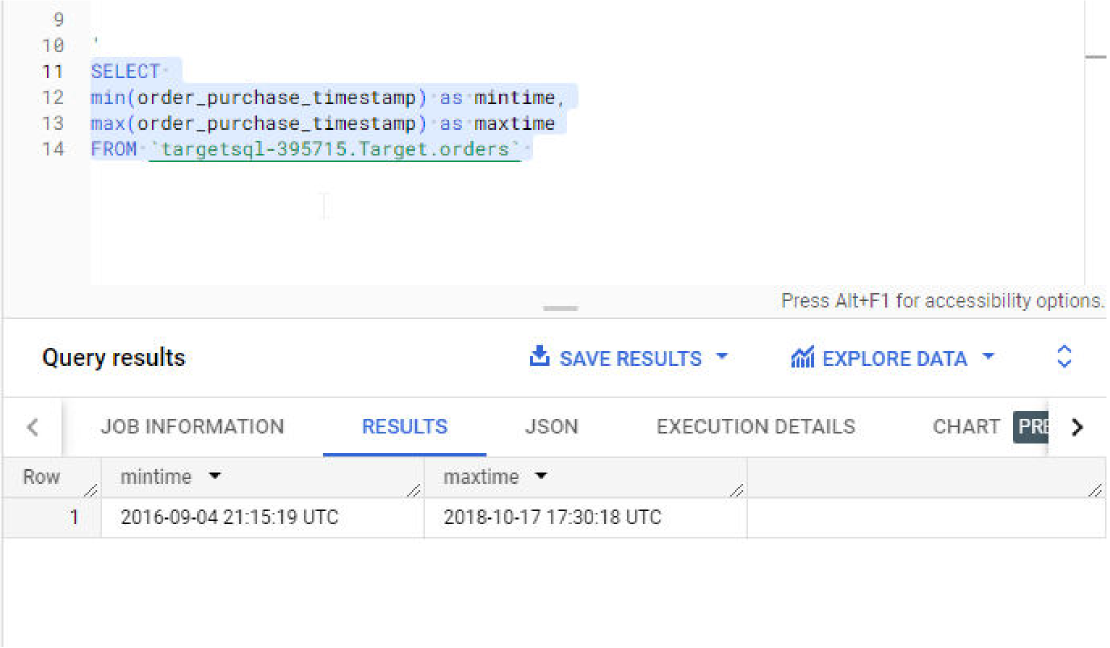
Q.Data type of all columns in the "customers" table



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

We have string and integer data type with nullable mode.

Q. Get the time range between which the orders were placed.

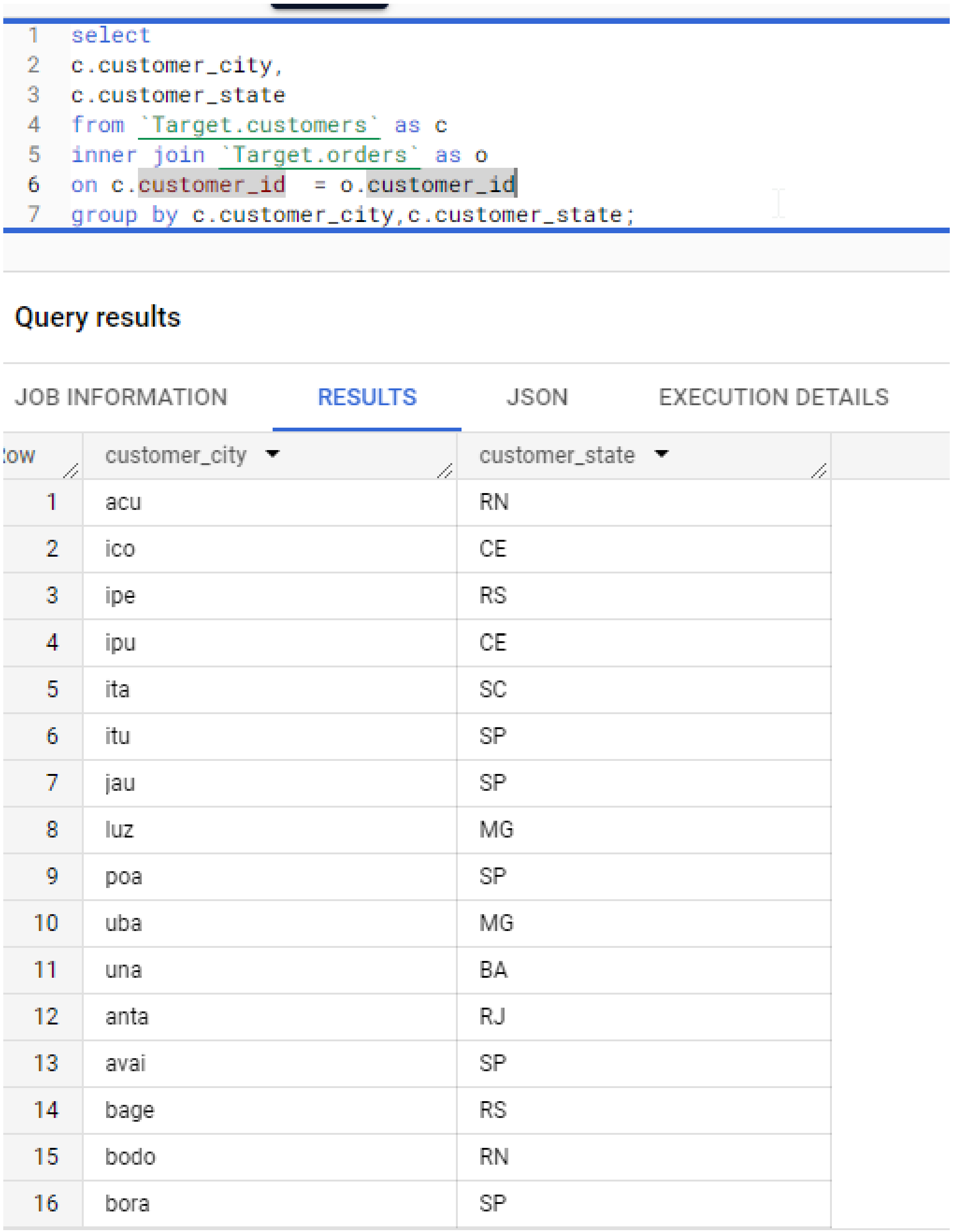


```
'
SELECT
min(order_purchase_timestamp) as mintime,
max(order_purchase_timestamp) as maxtime
FROM `targetsql-395715.Target.orders`
```

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

Insight - We have given data from 2016-09-04 to 2018-10-17

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

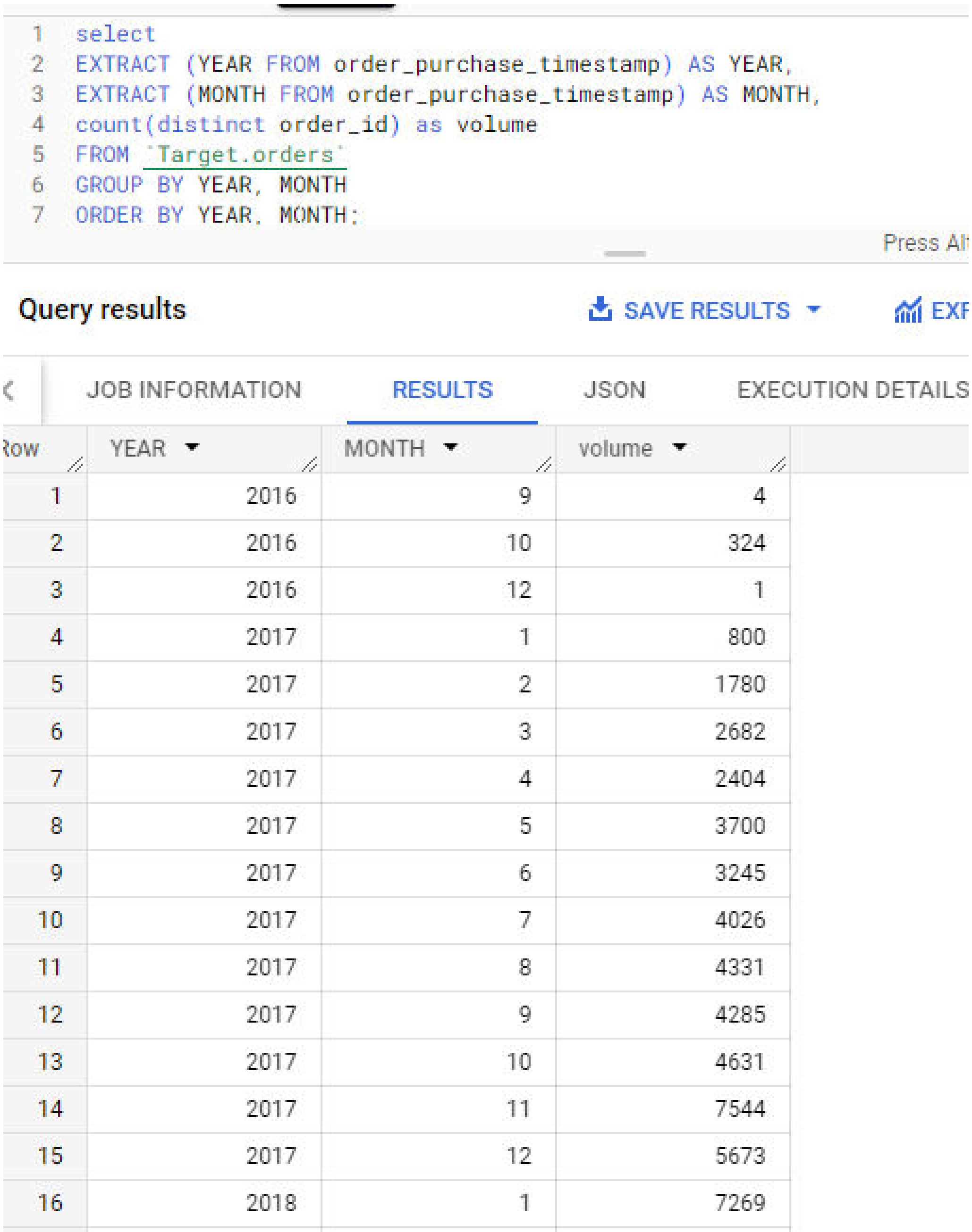


```
select
c.customer_city,
c.customer_state
from `Target.customers` as c
inner join `Target.orders` as o
on c.customer_id  = o.customer_id
group by c.customer_city,c.customer_state;
```

ow	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
12	anta	RJ
13	avai	SP
14	bage	RS
15	bodo	RN
16	bora	SP

Insight - We have 4310 states and cties of customer who ordered during the given period

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



```
select
EXTRACT (YEAR FROM order_purchase_timestamp) AS YEAR,
EXTRACT (MONTH FROM order_purchase_timestamp) AS MONTH,
count(distinct order_id) as volume
FROM `Target.orders`
GROUP BY YEAR, MONTH
ORDER BY YEAR, MONTH;
```

Row	YEAR	MONTH	volume
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269

insight - Analysis of ths observed trens reveals an upward trajectory between 2017 to 2018 but afterwards there has been notable downturn in performance with the target consistly experence a decline post-2018

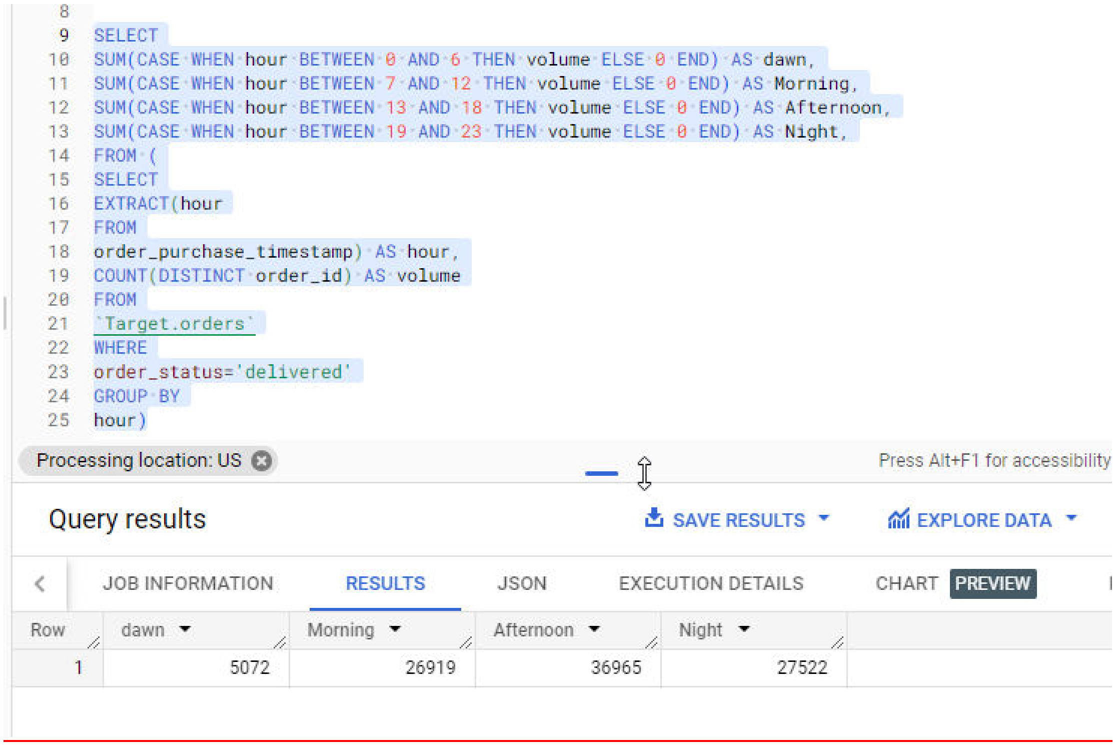
Q. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

0-6 hrs : Dawn

7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night



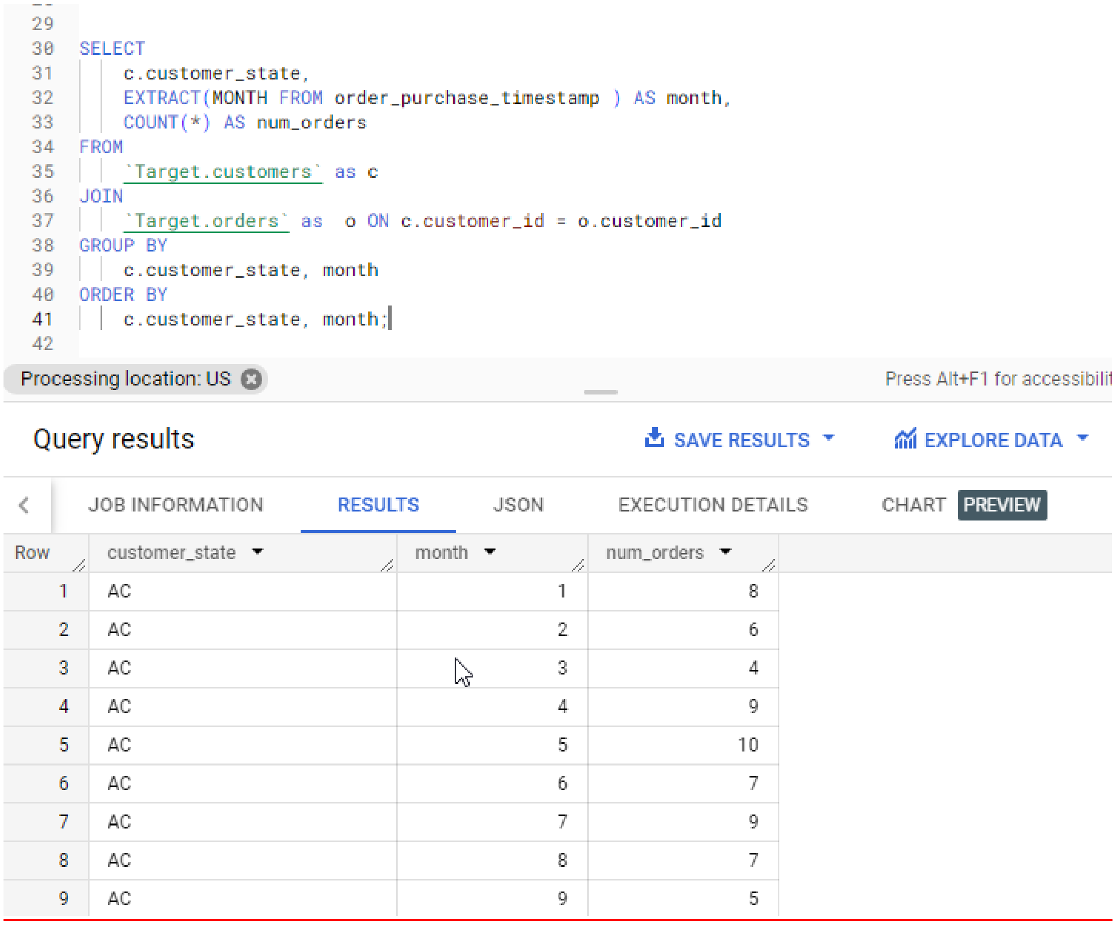
```
SELECT
SUM(CASE WHEN hour BETWEEN 0 AND 6 THEN volume ELSE 0 END) AS dawn,
SUM(CASE WHEN hour BETWEEN 7 AND 12 THEN volume ELSE 0 END) AS Morning,
SUM(CASE WHEN hour BETWEEN 13 AND 18 THEN volume ELSE 0 END) AS Afternoon,
SUM(CASE WHEN hour BETWEEN 19 AND 23 THEN volume ELSE 0 END) AS Night,
FROM (
SELECT
EXTRACT(hour
FROM
order_purchase_timestamp) AS hour,
COUNT(DISTINCT order_id) AS volume
FROM
`Target.orders`
WHERE
order_status='delivered'
GROUP BY
hour)
```

Row	dawn	Morning	Afternoon	Night
1	5072	26919	36965	27522

insight - After doing an analysis it mention clearly brazil people prefer their order placing at afternoon then night then morning and less no of order placed at dawn

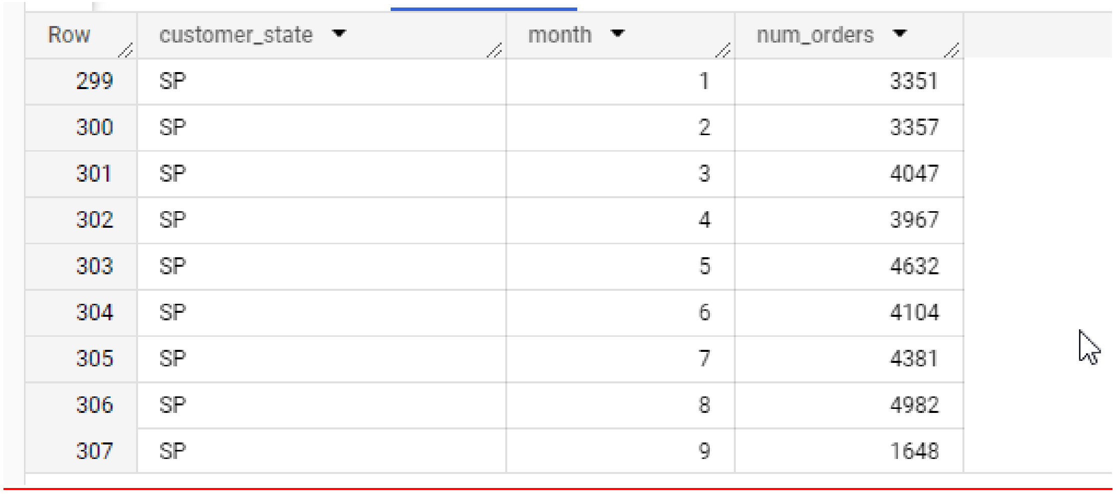
Recommendation - target create a discount offer at timing of 0 to 6 so more people can attract on that time and increase their sales at that time also.

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

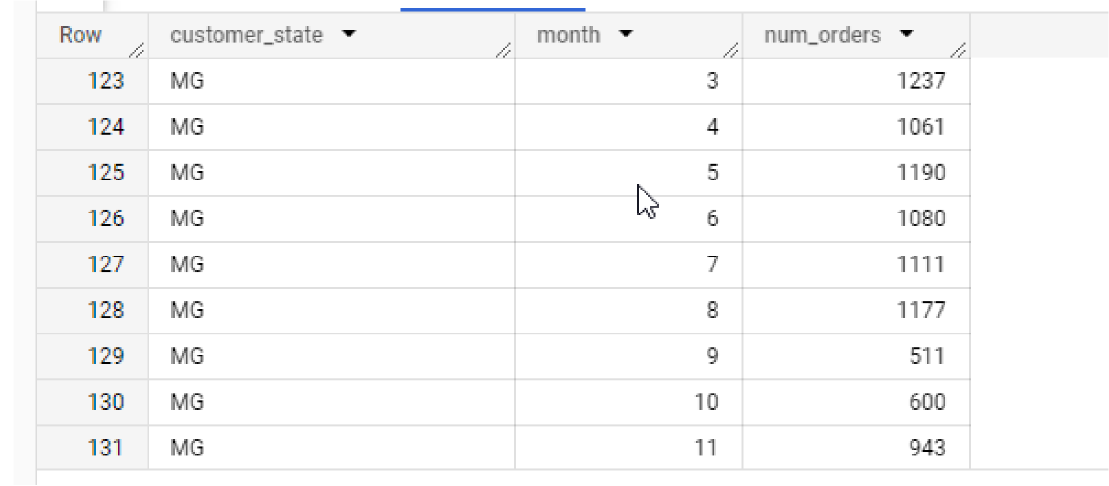


```sql
SELECT
    c.customer_state,
    EXTRACT(MONTH FROM order_purchase_timestamp ) AS month,
    COUNT(*) AS num_orders
FROM
    `Target.customers` as c
JOIN
    `Target.orders` as  o ON c.customer_id = o.customer_id
GROUP BY
    c.customer_state, month
ORDER BY
    c.customer_state, month;
```

Row	customer_state	month	num_orders
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



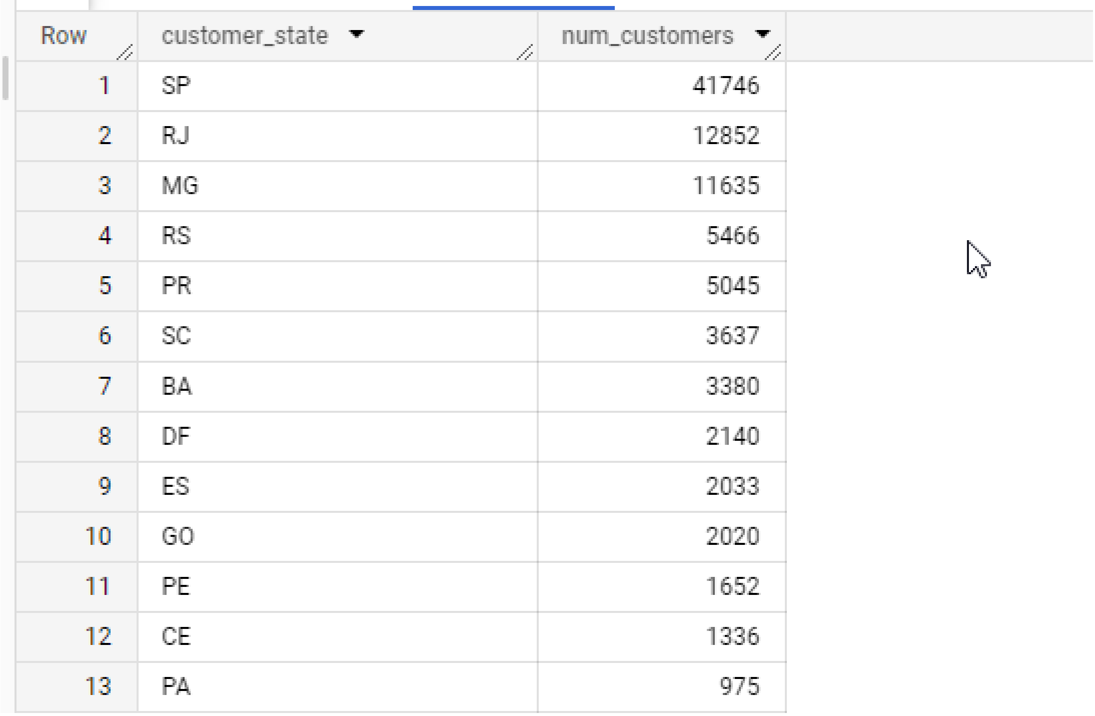
Row	customer_state	month	num_orders
299	SP	1	3351
300	SP	2	3357
301	SP	3	4047
302	SP	4	3967
303	SP	5	4632
304	SP	6	4104
305	SP	7	4381
306	SP	8	4982
307	SP	9	1648



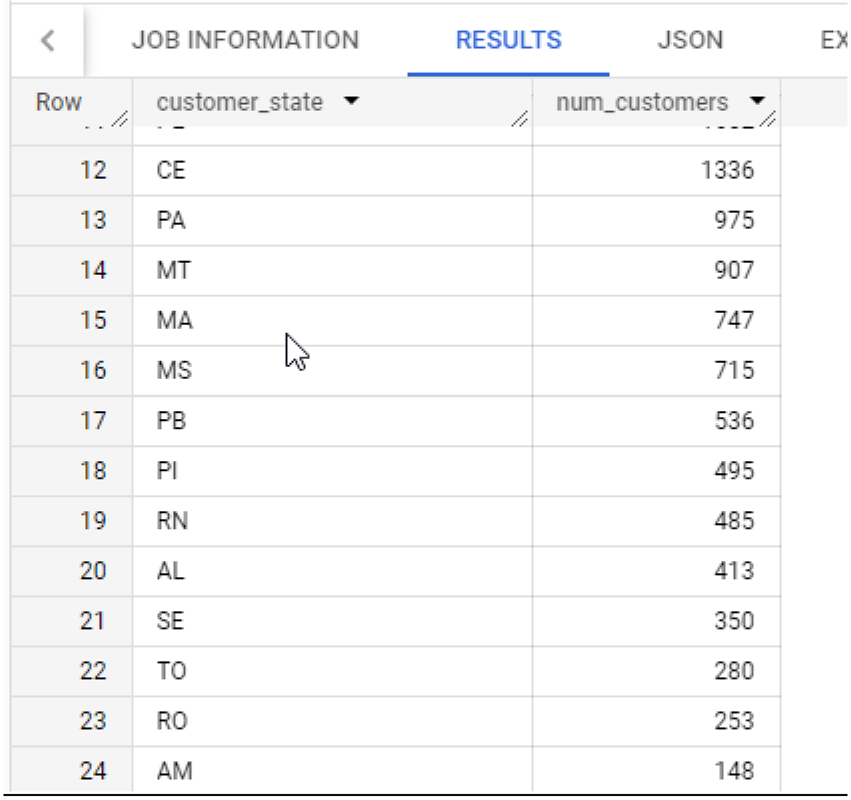
Row	customer_state	month	num_orders
123	MG	3	1237
124	MG	4	1061
125	MG	5	1190
126	MG	6	1080
127	MG	7	1111
128	MG	8	1177
129	MG	9	511
130	MG	10	600
131	MG	11	943

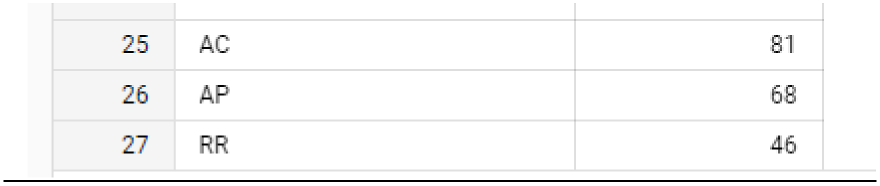
insight - After doing an analysis it shows cleary sp and mg states have placed their order highest on july and august

How are the customers distributed across all the states?



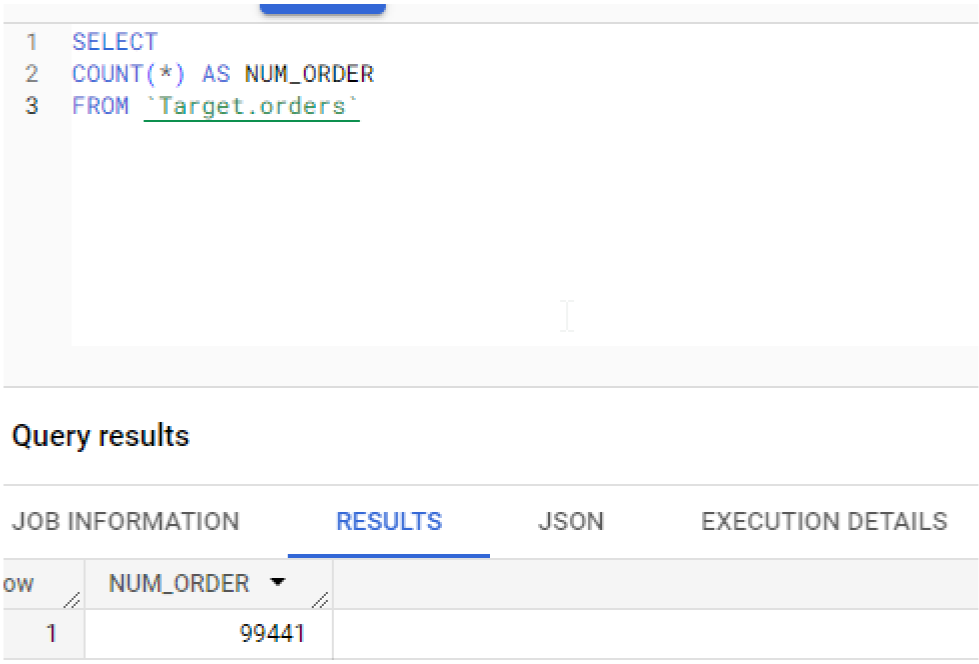
Row	customer_state	num_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336
13	PA	975





Row	customer_state	num_customers
12	CE	1336
13	PA	975
14	MT	907
15	MA	747
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

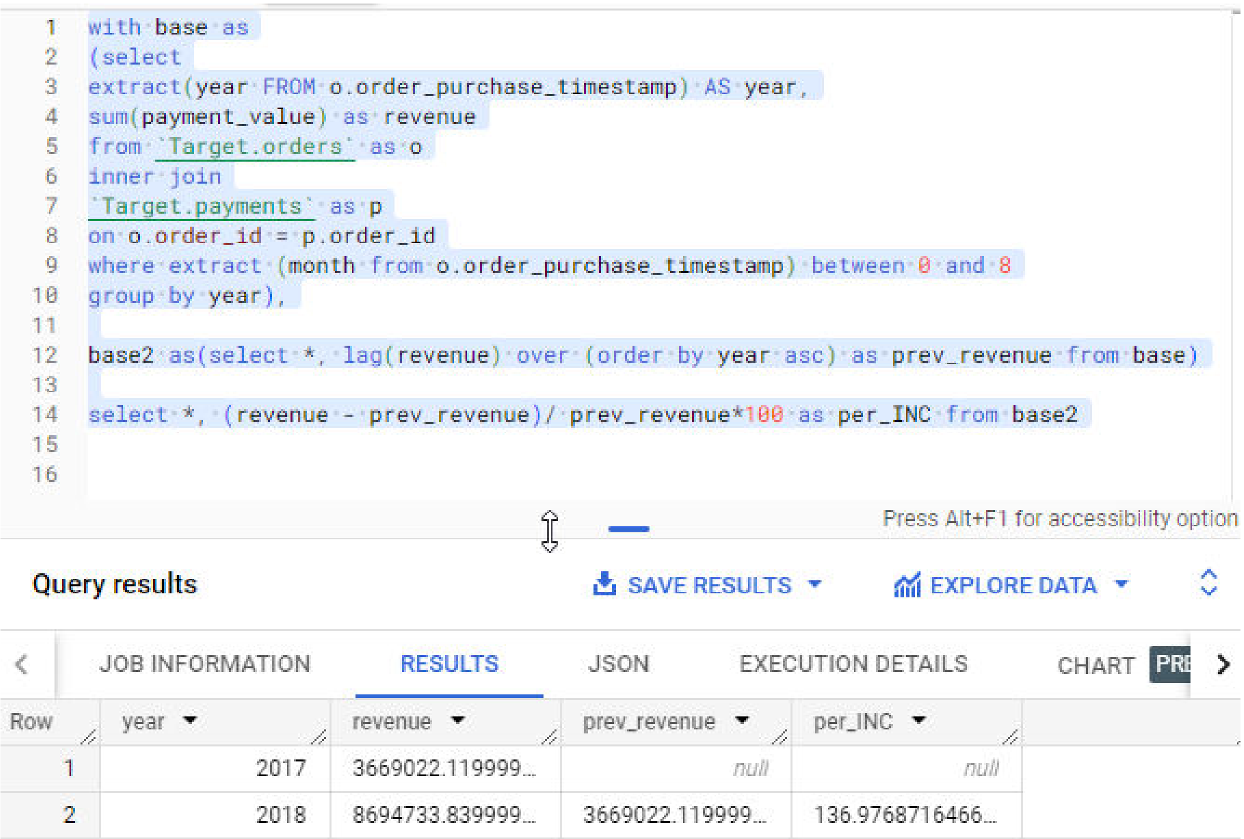
Above I HAVE TRIED TO LIST THE CUSTOMER DISTRIBUTED ACROSS ALL THE STATE



```
SELECT
COUNT(*) AS NUM_ORDER
FROM `Target.orders`
```

ow	NUM_ORDER
1	99441

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



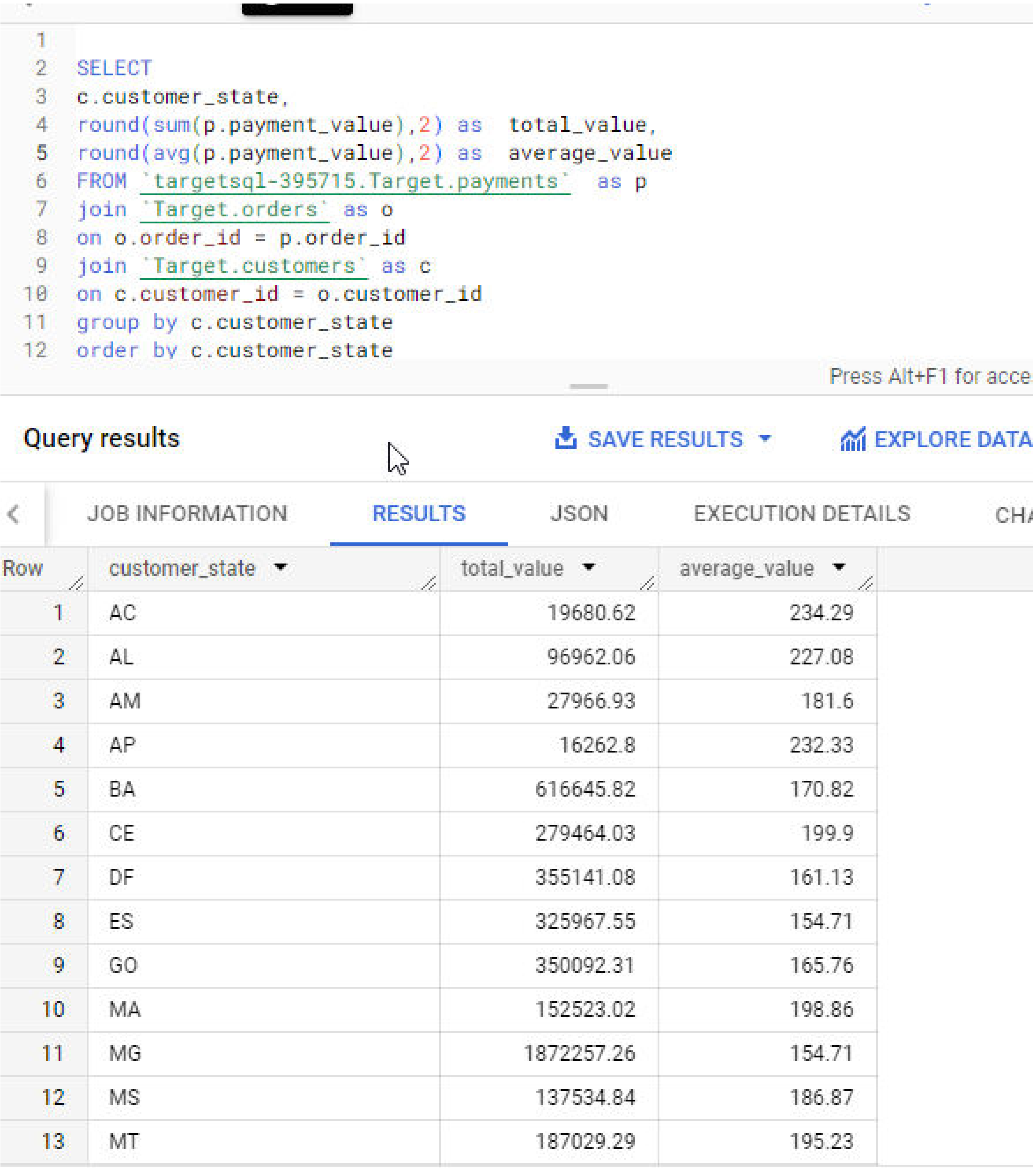
```
with base as
(select
extract(year FROM o.order_purchase_timestamp) AS year,
sum(payment_value) as revenue
from `Target.orders` as o
inner join
`Target.payments` as p
on o.order_id = p.order_id
where extract (month from o.order_purchase_timestamp) between 0 and 8
group by year),

base2 as(select *, lag(revenue) over (order by year asc) as prev_revenue from base)

select *, (revenue - prev_revenue)/ prev_revenue*100 as per_INC from base2
```

Row	year	revenue	prev_revenue	per_INC
1	2017	3669022.119999...	null	null
2	2018	8694733.839999...	3669022.119999...	136.9768716466...

insight - After doing an analysis it shows there is 136% growth in between 2017 and 2018.

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

Below is the list of total and average value of order price.

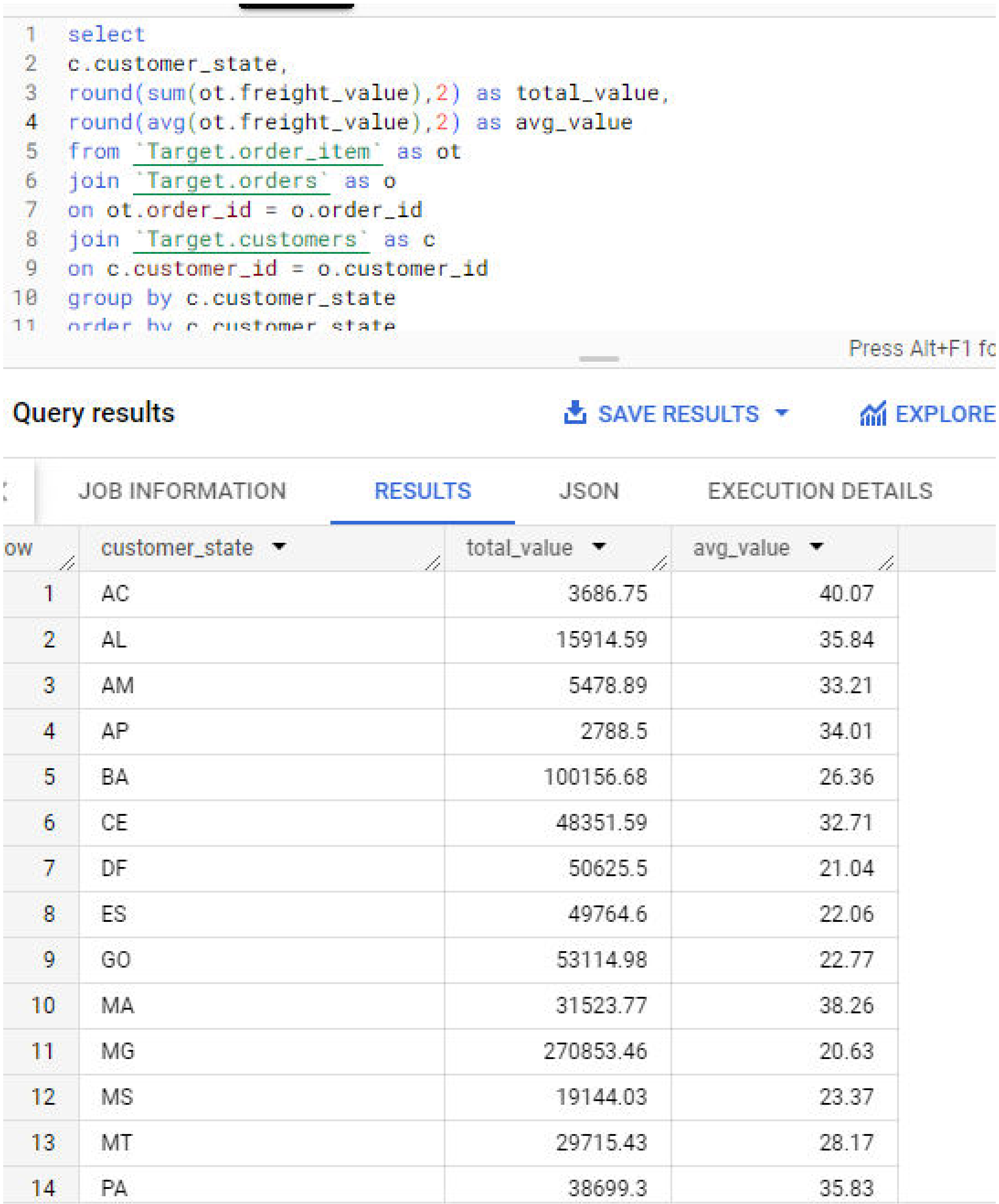


```

SELECT
c.customer_state,
round(sum(p.payment_value),2) as  total_value,
round(avg(p.payment_value),2) as  average_value
FROM `targetsql-395715.Target.payments`  as p
join `Target.orders` as o
on o.order_id = p.order_id
join `Target.customers` as c
on c.customer_id = o.customer_id
group by c.customer_state
order by c.customer_state
```

Row	customer_state	total_value	average_value
1	AC	19680.62	234.29
2	AL	96962.06	227.08
3	AM	27966.93	181.6
4	AP	16262.8	232.33
5	BA	616645.82	170.82
6	CE	279464.03	199.9
7	DF	355141.08	161.13
8	ES	325967.55	154.71
9	GO	350092.31	165.76
10	MA	152523.02	198.86
11	MG	1872257.26	154.71
12	MS	137534.84	186.87
13	MT	187029.29	195.23

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

Below I have try to find the all average value and total freight value



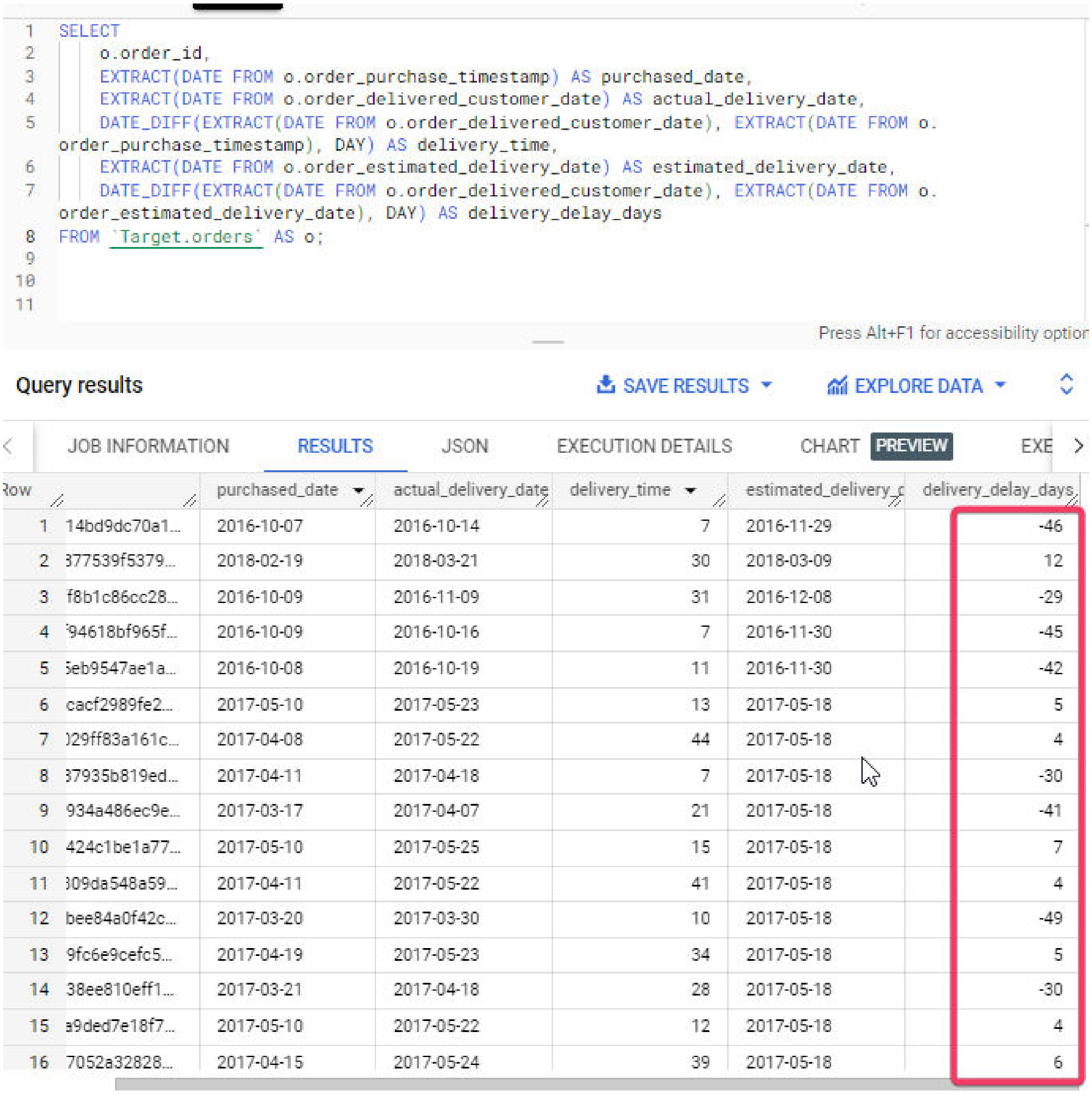
```
select
c.customer_state,
round(sum(ot.freight_value),2) as total_value,
round(avg(ot.freight_value),2) as avg_value
from `Target.order_item` as ot
join `Target.orders` as o
on ot.order_id = o.order_id
join `Target.customers` as c
on c.customer_id = o.customer_id
group by c.customer_state
order by c.customer_state
```

ow	customer_state	total_value	avg_value
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
11	MG	270853.46	20.63
12	MS	19144.03	23.37
13	MT	29715.43	28.17
14	PA	38699.3	35.83

Recommendation- If target can decrease the freight value then target can achieve better then its competent in terms of customer satisfaction- If freight value will decrease then it will decrease shipping charge so customer can get better benefit.

Demand increase – If shipping charge will low so product cost will automatically low, As compare to their competitor so demand will automatic increase for target business.

Q. 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.



```sql
SELECT
    o.order_id,
    EXTRACT(DATE FROM o.order_purchase_timestamp) AS purchased_date,
    EXTRACT(DATE FROM o.order_delivered_customer_date) AS actual_delivery_date,
    DATE_DIFF(EXTRACT(DATE FROM o.order_delivered_customer_date), EXTRACT(DATE FROM o.
order_purchase_timestamp), DAY) AS delivery_time,
    EXTRACT(DATE FROM o.order_estimated_delivery_date) AS estimated_delivery_date,
    DATE_DIFF(EXTRACT(DATE FROM o.order_delivered_customer_date), EXTRACT(DATE FROM o.
order_estimated_delivery_date), DAY) AS delivery_delay_days
FROM `Target.orders` AS o;
```

Row		purchased_date	actual_delivery_date	delivery_time	estimated_delivery_c	delivery_delay_days
1	14bd9dc70a1...	2016-10-07	2016-10-14	7	2016-11-29	-46
2	377539f5379...	2018-02-19	2018-03-21	30	2018-03-09	12
3	f8b1c86cc28...	2016-10-09	2016-11-09	31	2016-12-08	-29
4	94618bf965f...	2016-10-09	2016-10-16	7	2016-11-30	-45
5	5eb9547ae1a...	2016-10-08	2016-10-19	11	2016-11-30	-42
6	cacf2989fe2...	2017-05-10	2017-05-23	13	2017-05-18	5
7	029ff83a161c...	2017-04-08	2017-05-22	44	2017-05-18	4
8	37935b819ed...	2017-04-11	2017-04-18	7	2017-05-18	-30
9	934a486ec9e...	2017-03-17	2017-04-07	21	2017-05-18	-41
10	424c1be1a77...	2017-05-10	2017-05-25	15	2017-05-18	7
11	309da548a59...	2017-04-11	2017-05-22	41	2017-05-18	4
12	bee84a0f42c...	2017-03-20	2017-03-30	10	2017-05-18	-49
13	9fc6e9cefc5...	2017-04-19	2017-05-23	34	2017-05-18	5
14	38ee810eff1...	2017-03-21	2017-04-18	28	2017-05-18	-30
15	a9ded7e18f7...	2017-05-10	2017-05-22	12	2017-05-18	4
16	7052a32828...	2017-04-15	2017-05-24	39	2017-05-18	6

Insight in above table I have provided the no of days taken from the order 'purchase date in which we are able to see that in some cases it is about a month taken to deliver.

Recommendation – If target can work on delivery because from above table we are able to see delivery delays which is basically show actual delivery and estimated delivery which is in negative so if target can work on delivery procedure and can make delivered more quickly they can compete with their competitor.