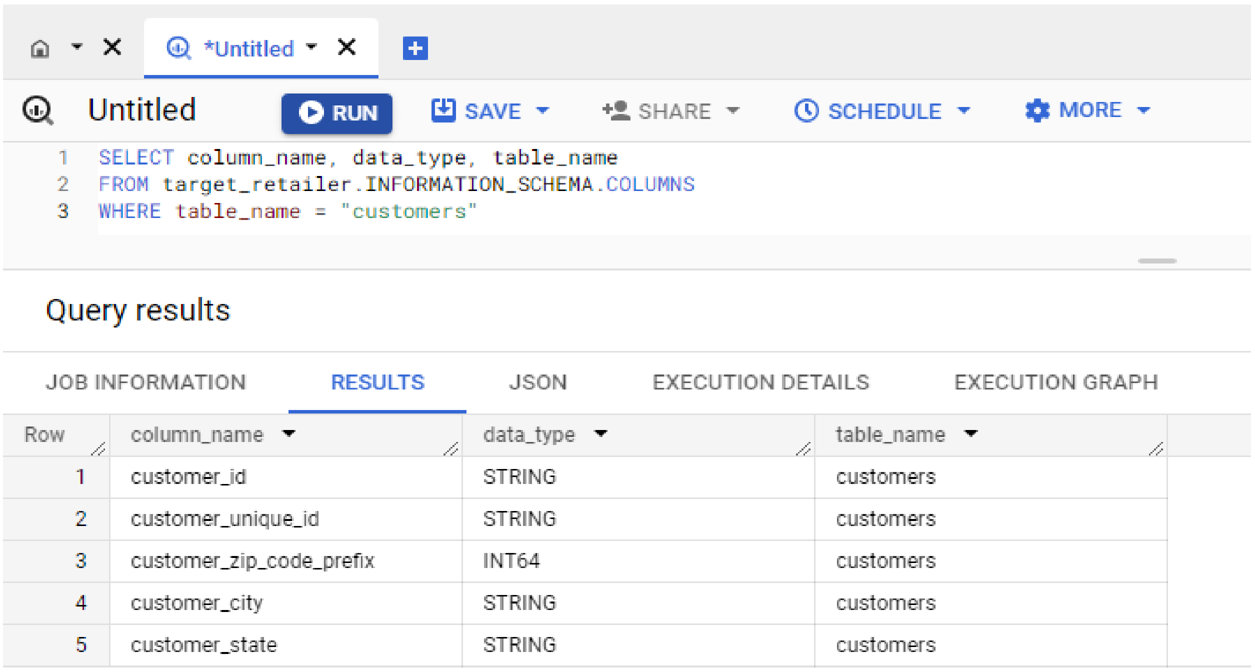
Business Case: Target SQL

Business Case Study - Company Target which is globally renowned brand and a prominent retailer in the United States. This particular business case focuses on the operations of Target in Brazil and provides insightful information -

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

QUERY :

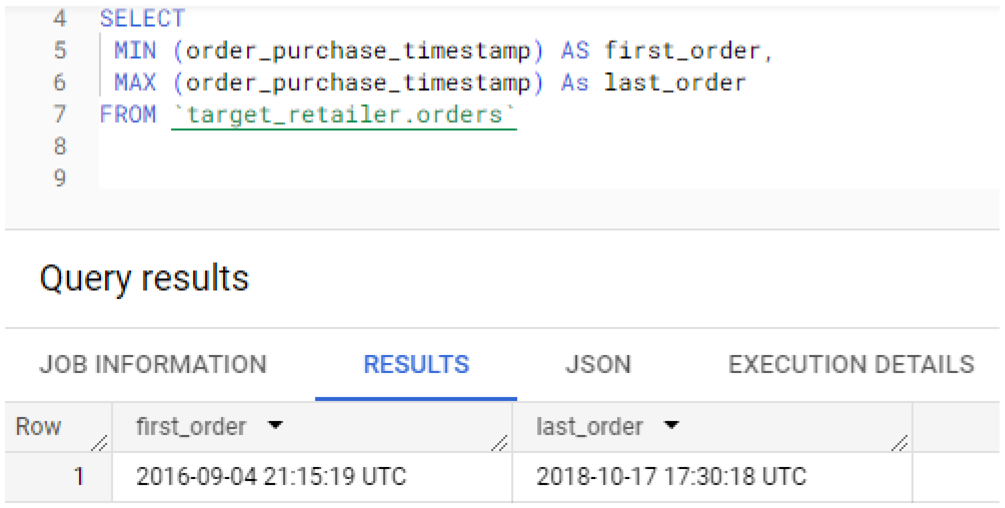


```sql
SELECT column_name, data_type, table_name
FROM target_retailer.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = "customers"
```

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

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

QUERY :



```sql
SELECT
 MIN (order_purchase_timestamp) AS first_order,
 MAX (order_purchase_timestamp) As last_order
FROM `target_retailer.orders`
```

Query results

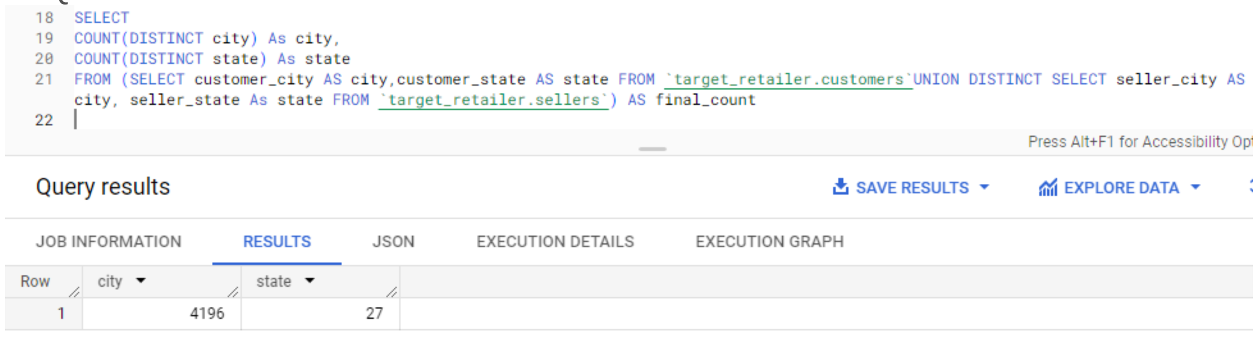
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

The first order placed is at 9:15 PM in 2016 and last order is placed at 5:30 PM in 2018.

1.3 Count the number of Cities and States in our dataset.

QUERY:



```sql
SELECT
COUNT(DISTINCT city) As city,
COUNT(DISTINCT state) As state
FROM (SELECT customer_city AS city,customer_state AS state FROM `target_retailer.customers`UNION DISTINCT SELECT seller_city AS
city, seller_state As state FROM `target_retailer.sellers`) AS final_count
```

Press Alt+F1 for Accessibility Opt

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	city	state
1	4196	27

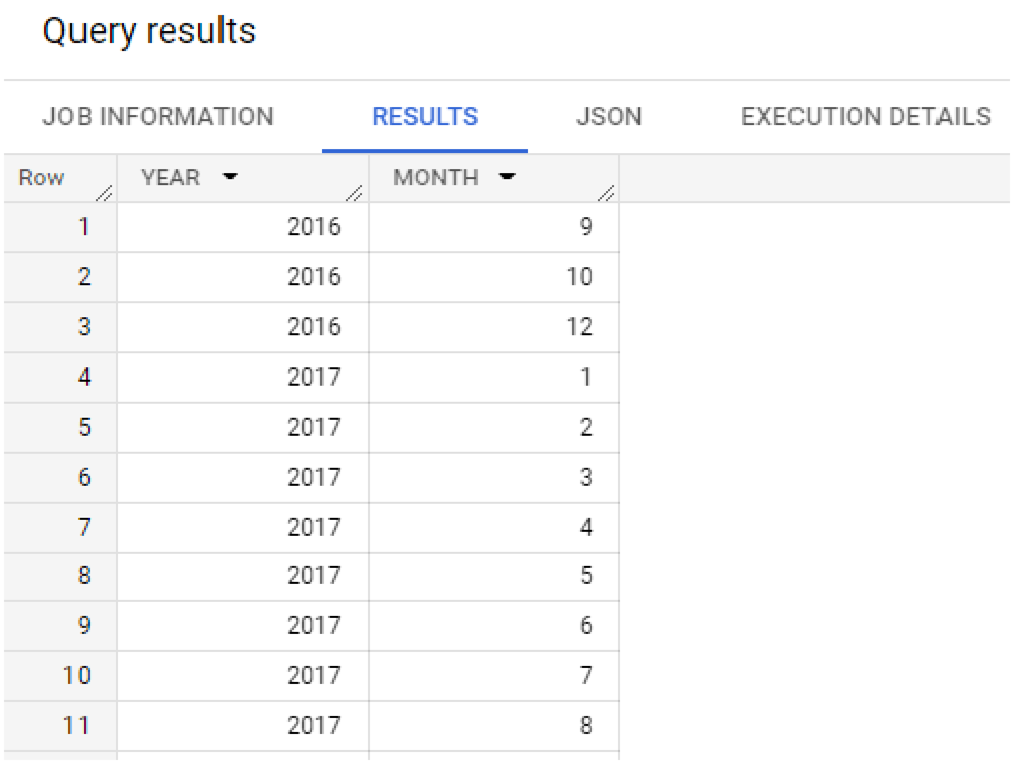
In dataset we can see here total cities are 4196 and states are 27 of Brazil.

2. In-depth Exploration

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

QUERY:

```
SELECT
EXTRACT (YEAR FROM order_purchase_timestamp) AS YEAR,
EXTRACT (MONTH FROM order_purchase_timestamp) AS MONTH
FROM `target_retailer.orders`
GROUP BY year,month
ORDER BY year,month
```



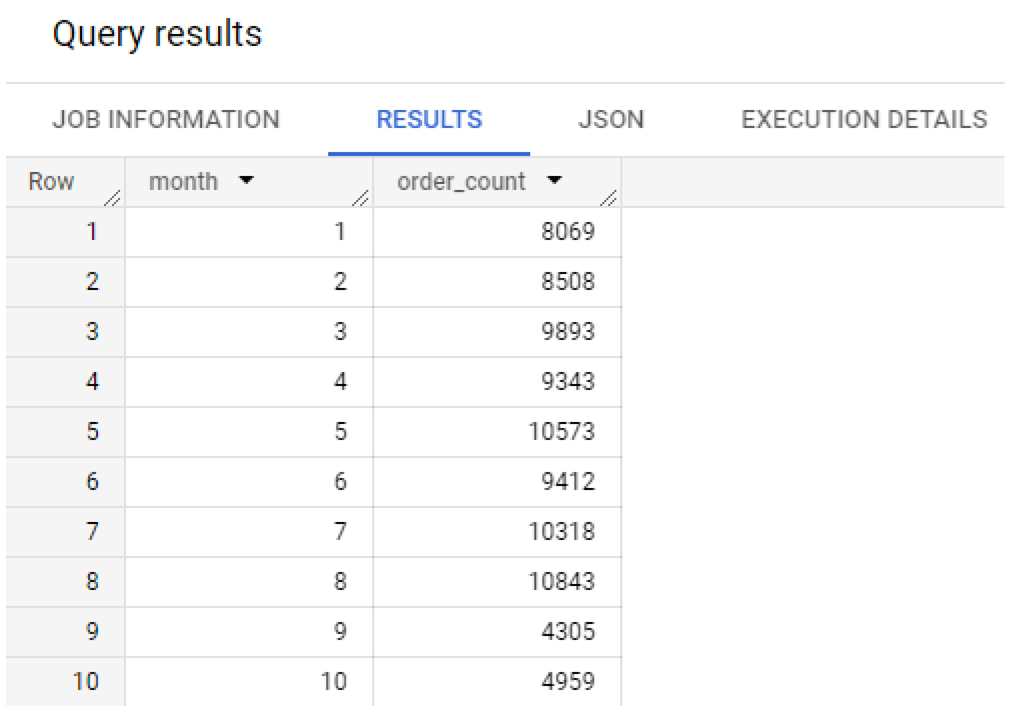
Row	YEAR	MONTH
1	2016	9
2	2016	10
3	2016	12
4	2017	1
5	2017	2
6	2017	3
7	2017	4
8	2017	5
9	2017	6
10	2017	7
11	2017	8

As we checked for number of orders it's not increasing there's constant change in number of orders sometimes it's increasing and sometime it's decreasing.

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

QUERY:

```
SELECT
EXTRACT (MONTH FROM order_purchase_timestamp) AS month,
COUNT (order_purchase_timestamp) AS order_count
FROM `target_retailer.orders`
GROUP BY month
ORDER BY month
```


Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	month	order_count
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959

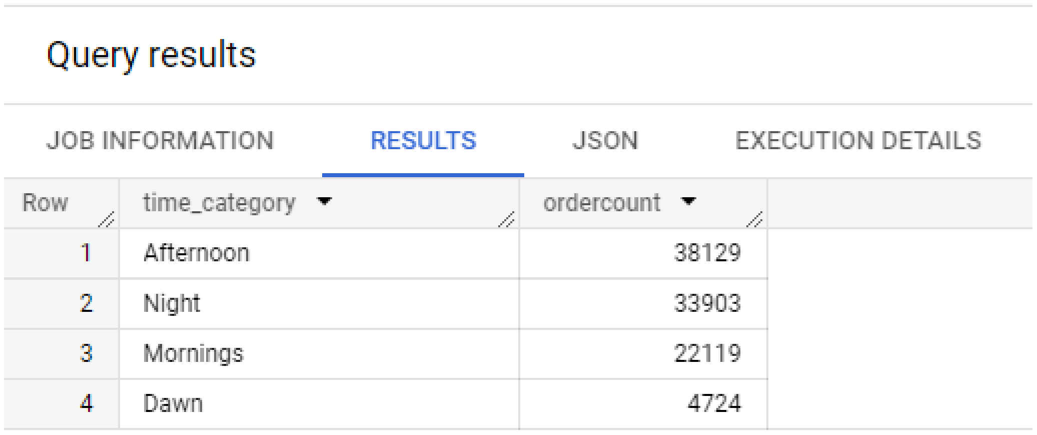
As we checked for change of orders monthwise the order count is highest in the month of August.

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

QUERY:

```
SELECT
a.time_category,
COUNT(a.count_order) AS ordercount
FROM (
SELECT
EXTRACT(HOUR FROM order_purchase_timestamp) AS hour,
COUNT(order_id) AS count_order,
CASE
WHEN EXTRACT(HOUR FROM order_purchase_timestamp)>= 0 AND EXTRACT (HOUR FROM
order_purchase_timestamp)< 6
THEN 'Dawn'
WHEN EXTRACT (HOUR FROM order_purchase_timestamp)>= 6 AND EXTRACT (HOUR FROM
order_purchase_timestamp)< 12
THEN 'Mornings'
WHEN EXTRACT (HOUR FROM order_purchase_timestamp)>= 12 AND EXTRACT (HOUR FROM
order_purchase_timestamp)< 18
THEN 'Afternoon'
WHEN EXTRACT (HOUR FROM order_purchase_timestamp)>= 18 AND EXTRACT (HOUR FROM
order_purchase_timestamp)<= 23
THEN 'Night'
END AS time_category FROM `target_retailer.orders`
GROUP BY
order_purchase_timestamp
) AS a
GROUP BY a.time_category
ORDER BY ordercount DESC
```


Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	time_category	ordercount
1	Afternoon	38129
2	Night	33903
3	Mornings	22119
4	Dawn	4724

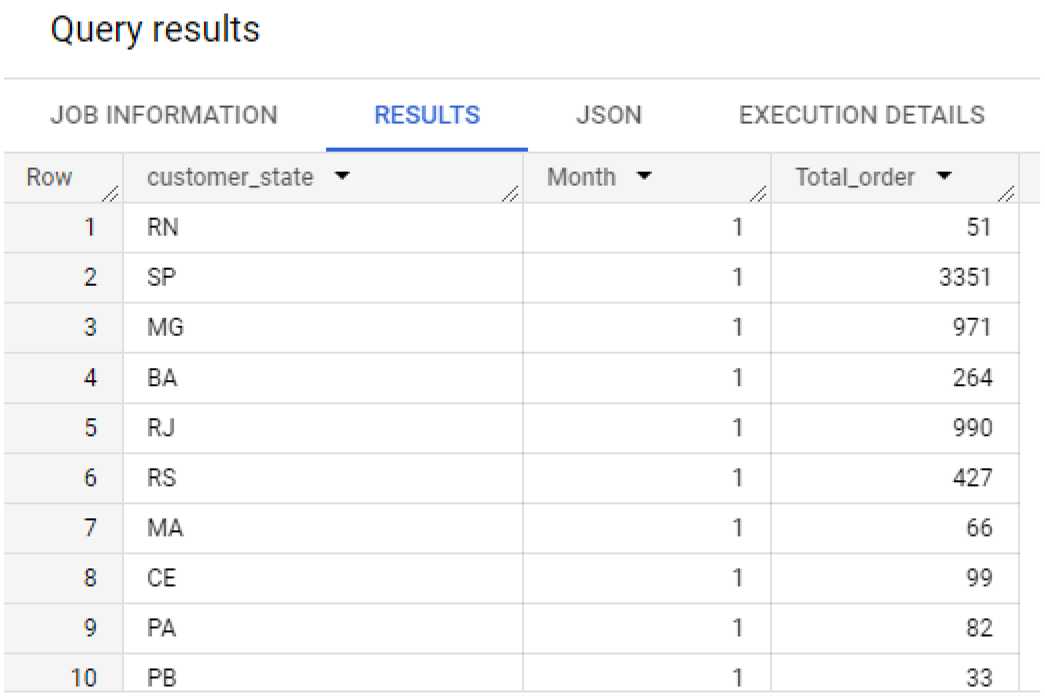
AS we cheked for order count for whole day, Brazilian customers tend to buy more in Afternoon time, so we can consider this as peak time.

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

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

QUERY:

```
SELECT c.customer_state,
EXTRACT (MONTH FROM o.order_purchase_timestamp) AS Month,COUNT(*) AS Total_order
FROM `target_retailer.customers`AS c JOIN `target_retailer.orders` AS o ON c.customer_id =
o.customer_id
GROUP BY Month,c.customer_state
ORDER BY Month
```

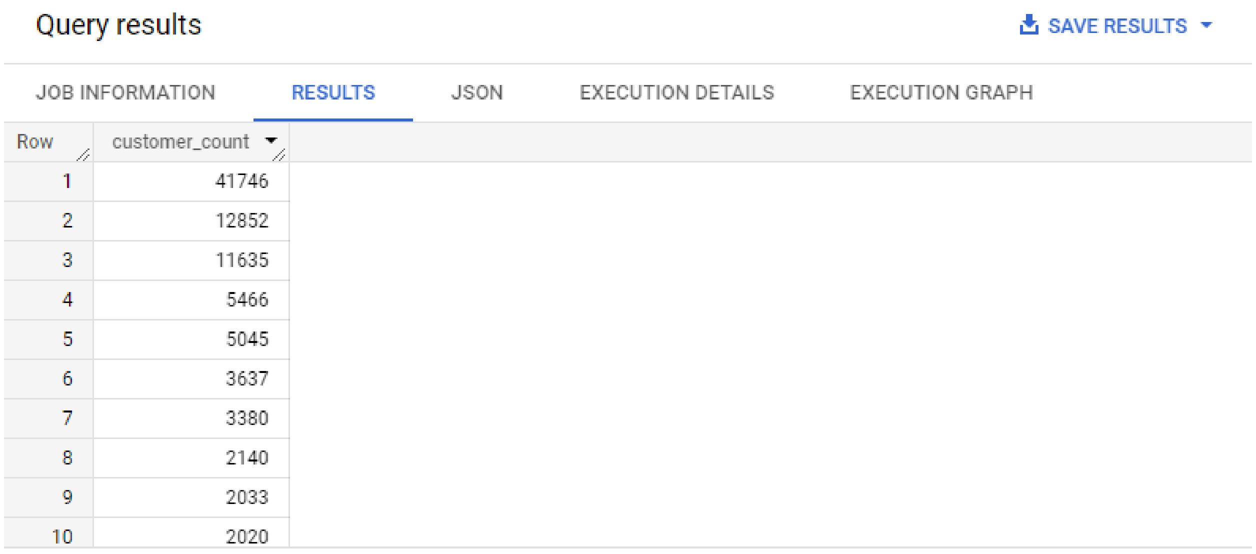

Query results

Row	customer_state	Month	Total_order
1	RN	1	51
2	SP	1	3351
3	MG	1	971
4	BA	1	264
5	RJ	1	990
6	RS	1	427
7	MA	1	66
8	CE	1	99
9	PA	1	82
10	PB	1	33

As we checked for total number of orders placed, different states are having different number of orders.

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

QUERY:
```
SELECT
COUNT (DISTINCT customer_id) AS customer_count
FROM `target_retailer.customers`
GROUP BY customer_state
ORDER BY customer_count DESC
```



Row	customer_count
1	41746
2	12852
3	11635
4	5466
5	5045
6	3637
7	3380
8	2140
9	2033
10	2020

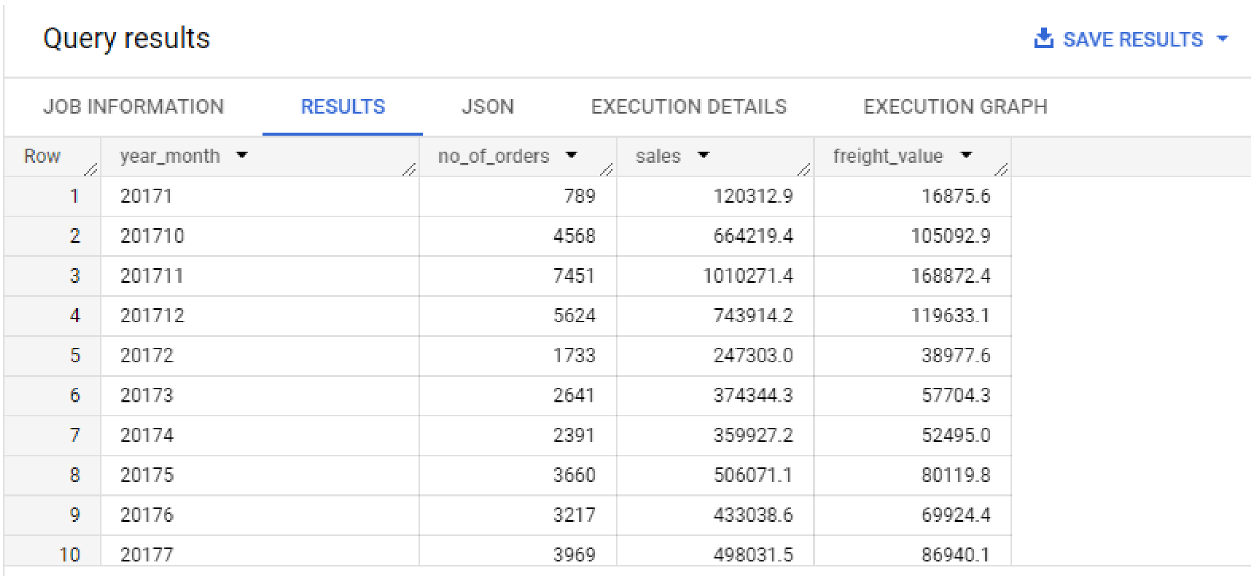
Here we can see the total count of customers in each state it’s varying state to state

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.

QUERY:

```
SELECT DISTINCT CONCAT(EXTRACT(YEAR
from O.order_purchase_timestamp),
EXTRACT (Month from
O.order_purchase_timestamp)) as year_month,
COUNT(distinct O.order_id) as no_of_orders,
ROUND(SUM(ol.price), 1) as sales,
ROUND(SUM(ol.freight_value), 1) as freight_value
FROM `target_retailer.orders` O
join `target_retailer.order_items` ol
on ol.order_id=O.order_id
GROUP BY year_month having
year_month >= '2017,1' and
year_month <= '2018, 8'
ORDER BY year_month;
```


Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

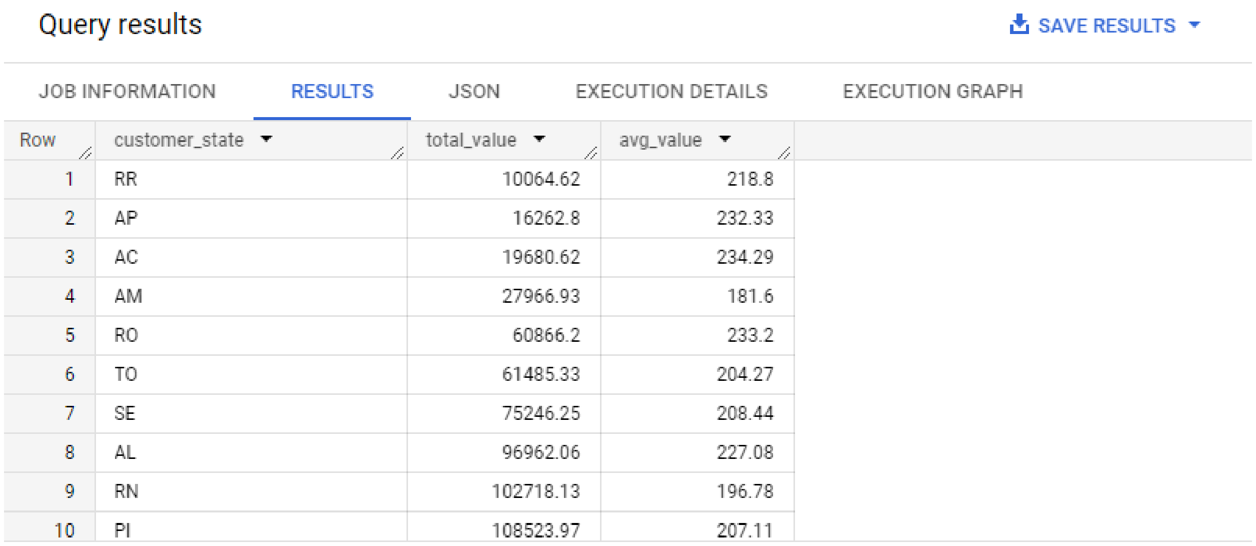
Row	year_month	no_of_orders	sales	freight_value
1	20171	789	120312.9	16875.6
2	201710	4568	664219.4	105092.9
3	201711	7451	1010271.4	168872.4
4	201712	5624	743914.2	119633.1
5	20172	1733	247303.0	38977.6
6	20173	2641	374344.3	57704.3
7	20174	2391	359927.2	52495.0
8	20175	3660	506071.1	80119.8
9	20176	3217	433038.6	69924.4
10	20177	3969	498031.5	86940.1

Here the cost of orders and sales both are increasing and decreasing every year.

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

QUERY:

```
SELECT DISTINCT CONCAT(EXTRACT(YEAR
from O.order_purchase_timestamp),
EXTRACT (Month from
O.order_purchase_timestamp)) as year_month,
COUNT(distinct O.order_id) as no_of_orders,
ROUND(SUM(ol.price), 1) as sales,
ROUND(SUM(ol.freight_value), 1) as freight_value
FROM `target_retailer.orders` O
join `target_retailer.order_items` ol
on ol.order_id=O.order_id
GROUP BY year_month having
year_month >= '2017,1' and
year_month <= '2018, 8'
ORDER BY year_month;
```


Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

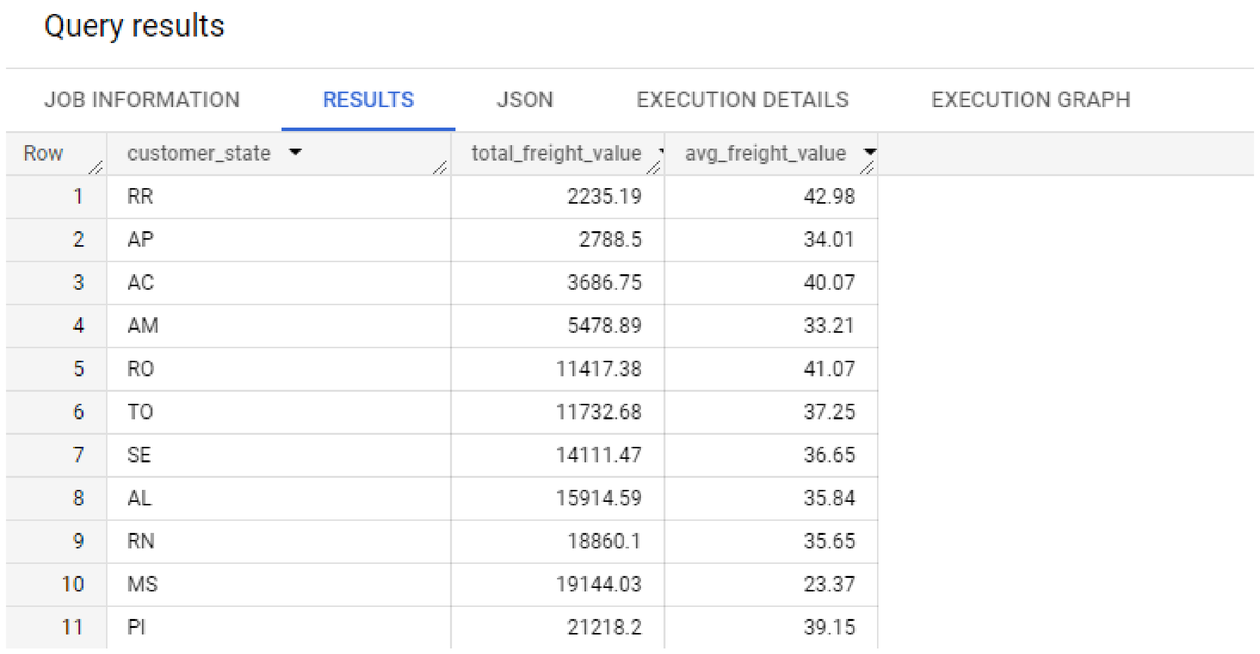
Row	customer_state	total_value	avg_value
1	RR	10064.62	218.8
2	AP	16262.8	232.33
3	AC	19680.62	234.29
4	AM	27966.93	181.6
5	RO	60866.2	233.2
6	TO	61485.33	204.27
7	SE	75246.25	208.44
8	AL	96962.06	227.08
9	RN	102718.13	196.78
10	PI	108523.97	207.11

Here we have checked for average value of order price of each state.

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

QUERY:

```
SELECT
c.customer_state,
ROUND (SUM(ot.freight_value),2) AS total_freight_value,
ROUND (AVG(ot.freight_value),2) AS avg_freight_value
FROM `target_retailer.order_items` AS ot JOIN `target_retailer.orders` AS o ON ot.order_id =
o.order_id
JOIN `target_retailer.customers` AS c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY total_freight_value,
avg_freight_value
```



Row	customer_state	total_freight_value	avg_freight_value
1	RR	2235.19	42.98
2	AP	2788.5	34.01
3	AC	3686.75	40.07
4	AM	5478.89	33.21
5	RO	11417.38	41.07
6	TO	11732.68	37.25
7	SE	14111.47	36.65
8	AL	15914.59	35.84
9	RN	18860.1	35.65
10	MS	19144.03	23.37
11	PI	21218.2	39.15

Here we have checked for total and average value of order freight for each state different states having different values.

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

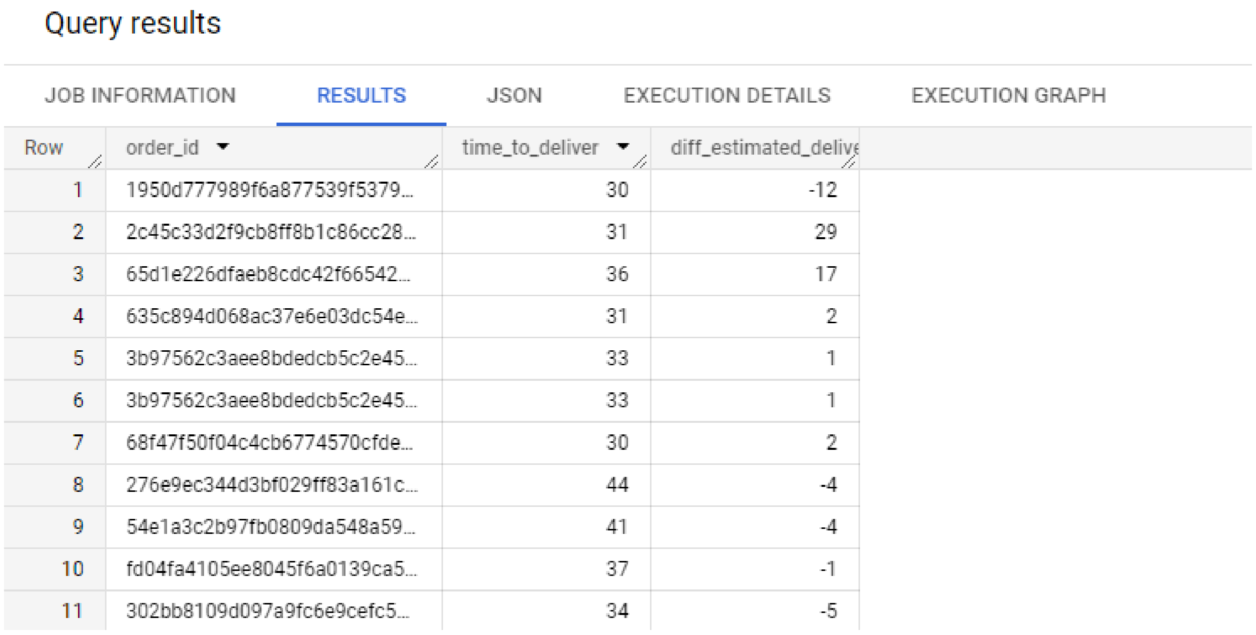
5.1 Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

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

QUERY:

```
SELECT
o.order_id,
EXTRACT(DAY FROM(date(o.order_delivered_customer_date)- date(o.order_purchase_timestamp))) As
time_to_deliver,
EXTRACT(DAY FROM(date(o.order_estimated_delivery_date)-
date(o.order_delivered_customer_date))) As diff_estimated_delivery
FROM `target_retailer.orders` As o
LEFT JOIN `target_retailer.order_items` As ot
ON ot.order_id = o. order_id
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

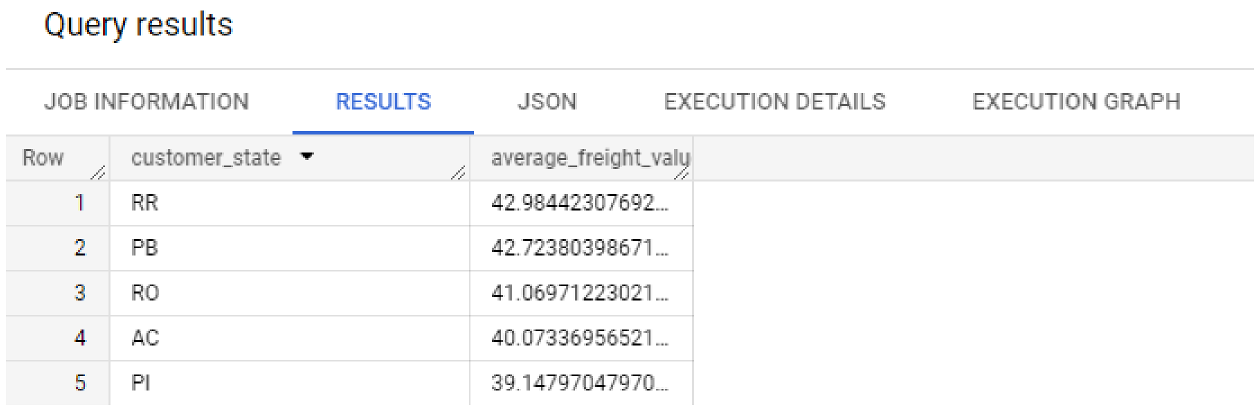
Row	order_id	time_to_deliver	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	31	29
3	65d1e226dfaeb8cdc42f66542...	36	17
4	635c894d068ac37e6e03dc54e...	31	2
5	3b97562c3aee8bdedcb5c2e45...	33	1
6	3b97562c3aee8bdedcb5c2e45...	33	1
7	68f47f50f04c4cb6774570cfde...	30	2
8	276e9ec344d3bf029ff83a161c...	44	-4
9	54e1a3c2b97fb0809da548a59...	41	-4
10	fd04fa4105ee8045f6a0139ca5...	37	-1
11	302bb8109d097a9fc6e9cefc5...	34	-5

Here we can see the orders mostly have been delayed for some days and not been delivered at the estimated date.

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

QUERY:

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS average_freight_value
FROM `target_retailer.customers` AS c
JOIN `target_retailer.orders` AS o
ON c.customer_id = o.customer_id
JOIN `target_retailer.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY average_freight_value DESC
LIMIT 5;
```



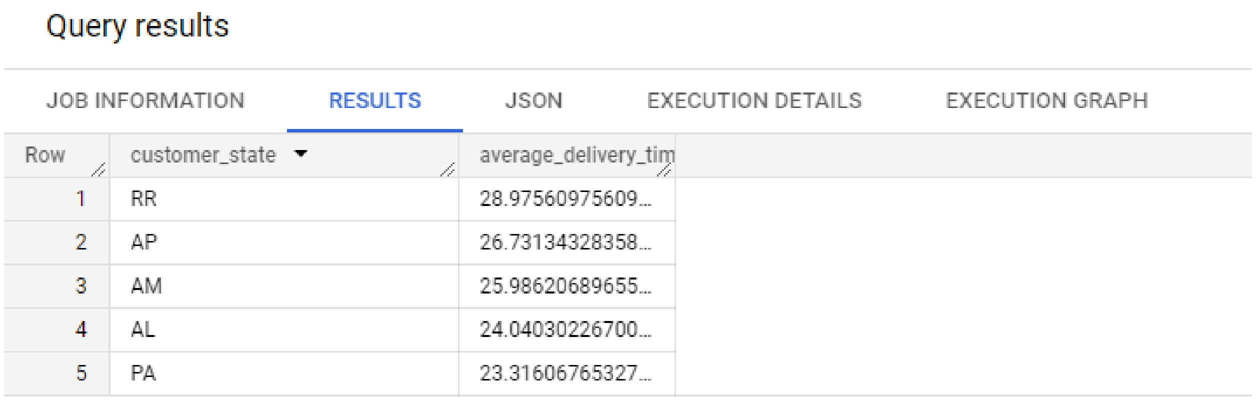
Row	customer_state	average_freight_valu
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

As we checked average freight rate for top 5 states, the state PI is having the lowest average freight value and state RR is having the highest average freight value.

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

QUERY:

```
SELECT
c.customer_state,
AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY))
AS average_delivery_time
FROM `target_retailer.customers`AS c
JOIN `target_retailer.orders` As o
ON c.customer_id = o.customer_id
WHERE o.order_delivered_customer_date IS NOT NULL
AND o.order_purchase_timestamp IS NOT NULL
GROUP BY c.customer_state
ORDER BY average_delivery_time DESC
LIMIT 5;
```


Query results

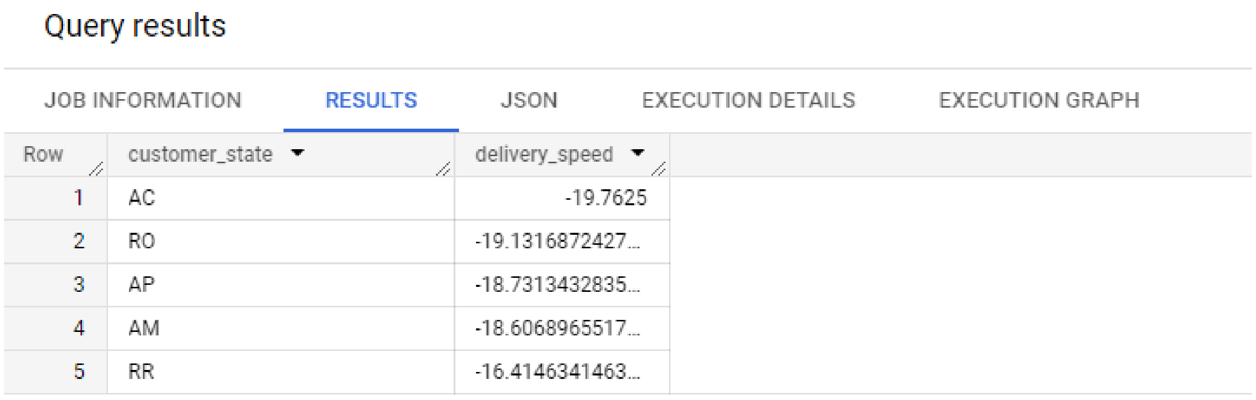
Row	customer_state	average_delivery_tim
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...

As we checked average delivery time for top 5 states, the state PA is having the lowest average delivery time and state RR is having the highest 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.

QUERY:

```
SELECT
c.customer_state,
AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date,
DAY)) AS delivery_speed
FROM `target_retailer.customers` AS c
JOIN `target_retailer.orders` AS o
ON c.customer_id = o.customer_id
GROUP BY c.customer_state
HAVING delivery_speed < 0
ORDER BY delivery_speed ASC
LIMIT 5;
```


Query results

Row	customer_state	delivery_speed
1	AC	-19.7625
2	RO	-19.1316872427...
3	AP	-18.7313432835...
4	AM	-18.6068965517...
5	RR	-16.4146341463...

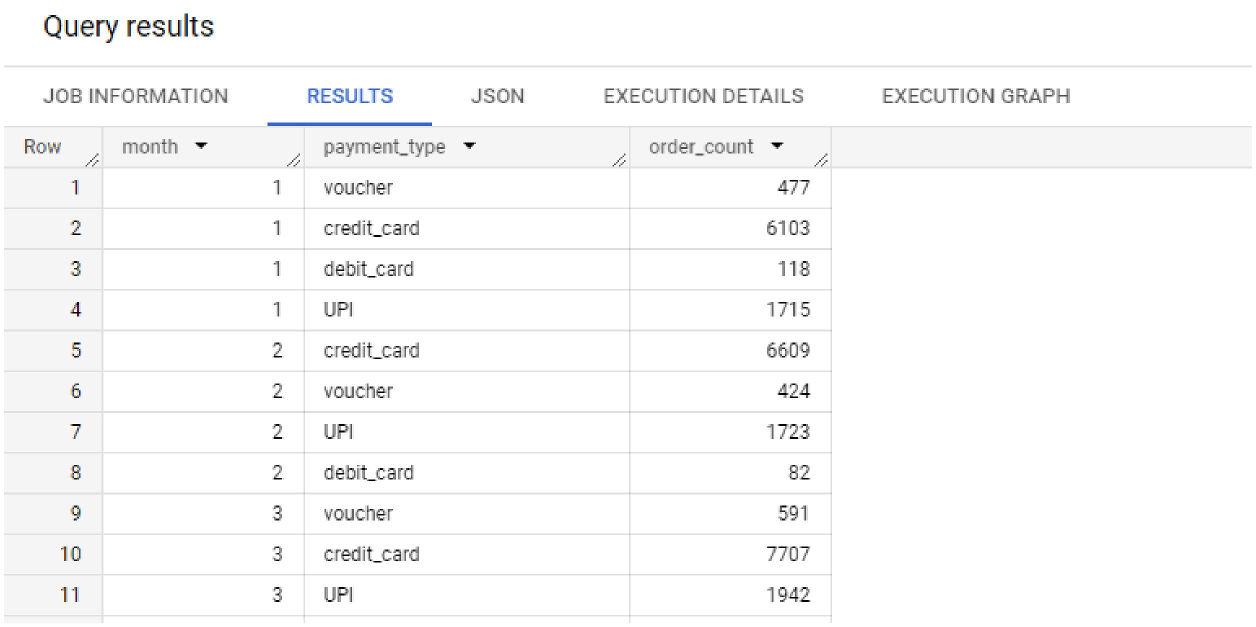
As we checked for the actual delivery date, not a single state had received delivery before estimated delivery date.

6. Analysis based on the payments:

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

QUERY:

```
SELECT
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
p.payment_type,
COUNT(*) AS order_count
FROM `target_retailer.orders` AS o
JOIN `target_retailer.payments` AS p
ON o.order_id = p.order_id
GROUP BY month, p.payment_type
ORDER BY month ASC;
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

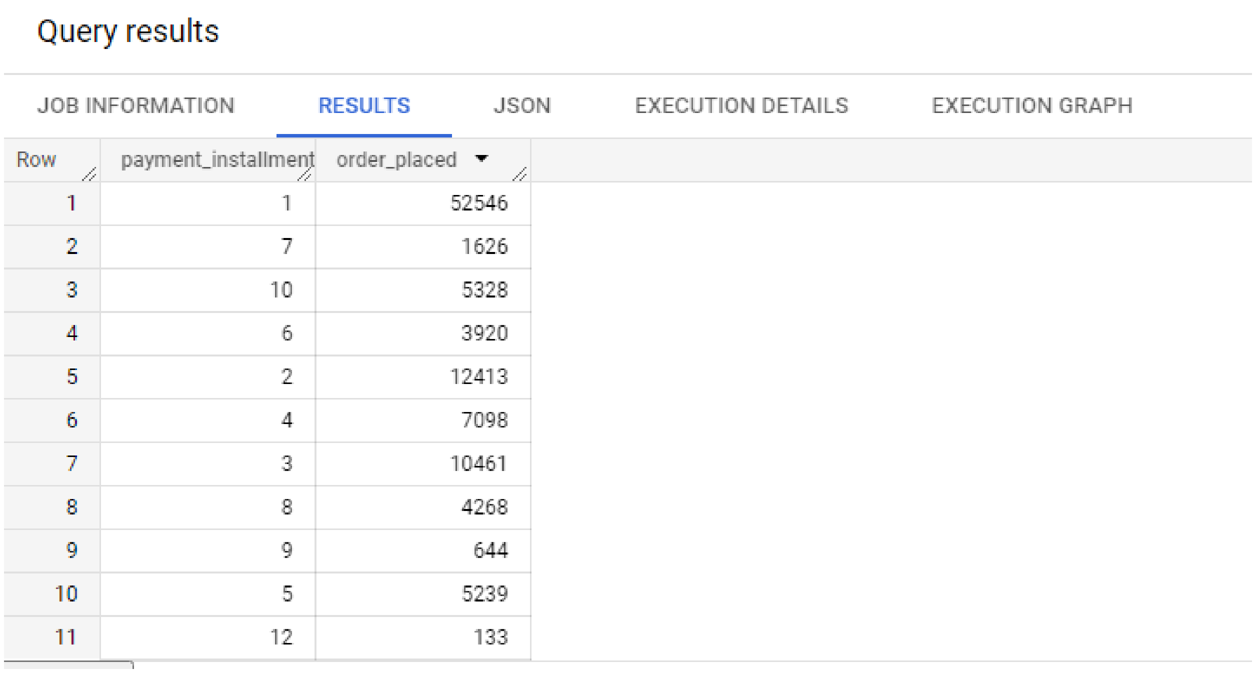
Row	month	payment_type	order_count
1	1	voucher	477
2	1	credit_card	6103
3	1	debit_card	118
4	1	UPI	1715
5	2	credit_card	6609
6	2	voucher	424
7	2	UPI	1723
8	2	debit_card	82
9	3	voucher	591
10	3	credit_card	7707
11	3	UPI	1942

Here as we checked for mode of payment customers mostly making payment online through credit cards , debit cards , UPI , Vouchers etc.

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

QUERY:

```
SELECT
p.payment_installments,
COUNT (*) AS order_placed
FROM `target_retailer.orders` AS o
JOIN `target_retailer.payments` AS p
ON o.order_id = p.order_id
GROUP BY p.payment_installments;
```



Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	payment_installment	order_placed
1	1	52546
2	7	1626
3	10	5328
4	6	3920
5	2	12413
6	4	7098
7	3	10461
8	8	4268
9	9	644
10	5	5239
11	12	133

Here we have checked for number of orders which have been paid in installments.

RECOMMENDATION -

Through this business case study I came to know various things which I have mentioned below –

1. In past years we can see, higher number of orders were placed in rainy season. So company should keep seasonal and festive offers during other seasons.
2. In full day Brazilian customers buying alot in the Afternoon time, so in other time company can keep offers , give promo code and complimentary goodies so this type of strategies can be implemented to seek attention and increase their sales to make more profits.
3. In some states the company is not having more customers so company should advertise by promoting more they can reach out each and every state easily to seek attention.
4. Company is spending more on transport deliveries are not delivered on the estimated date here , company can open more branches in different states to reduce their cost of transport and

also the delivery will be done on the estimated date it will help company to build Good image in Market ultimately increase in sales will be seen.

5. Company can cut down on cost by buying raw material/products in bulk from distributors and offer Low rates of products in market, this will increase the revenue of company as the sales will scale up.