Task 6 at Elevate Labs

Dataset: Target Retail Store

Submission by

Nikhil Kumar Nigam

9215949494

Email: nikhilnigam@engineer.com

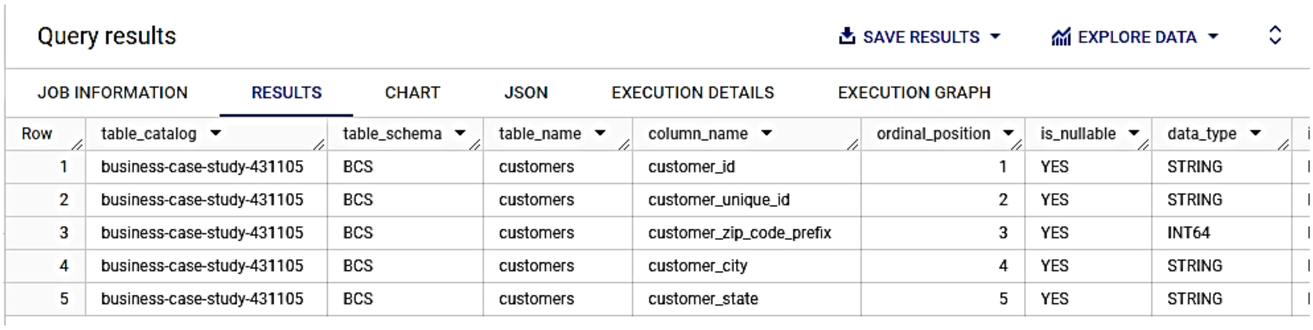
Er.nikhil2007@gmail.com

1. Importing the dataset and doing usual exploratory analysis steps like checking the structure & characteristics of the dataset:

a. Checking the data type of all columns in the "customers" table.

Query

```sql
select *, data_type
from  `business-case-study-431105.BCS.INFORMATION_SCHEMA.COLUMNS`
where  table_name = 'customers';
```



Row	table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type	i
1	business-case-study-431105	BCS	customers	customer_id	1	YES	STRING	I
2	business-case-study-431105	BCS	customers	customer_unique_id	2	YES	STRING	I
3	business-case-study-431105	BCS	customers	customer_zip_code_prefix	3	YES	INT64	I
4	business-case-study-431105	BCS	customers	customer_city	4	YES	STRING	I
5	business-case-study-431105	BCS	customers	customer_state	5	YES	STRING	I

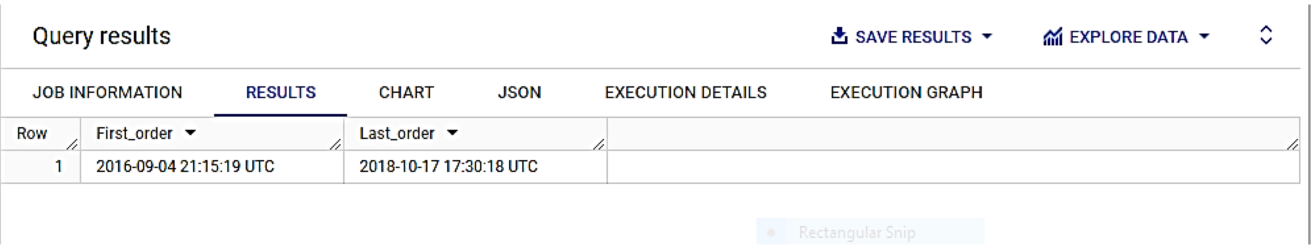
INFERENCE:

Customer_id, customer_unique_id, customer_city and customer_state are of STRING DATA TYPE where as customer_zip_code_prefix is of (INT64) INTEGER DATA TYPE.

b. Getting 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 BCS.orders
```



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

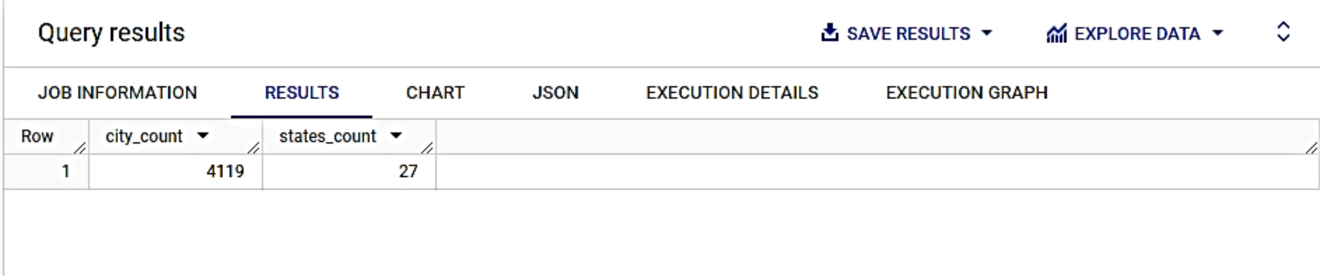
INFERENCE:

From the order table it can be analyzed that 'TARGET' received its first order on "2016-09-04" and Last order on "2018-10-17".

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

QUERY:

```
select
    count(distinct c.customer_city)  city_count,
    count(distinct c.customer_state)  states_count
from
    BCS.customers c
inner join
    BCS.orders o
on
    c.customer_id = o.customer_id;
```



Row	city_count	states_count
1	4119	27

INFERENCE :

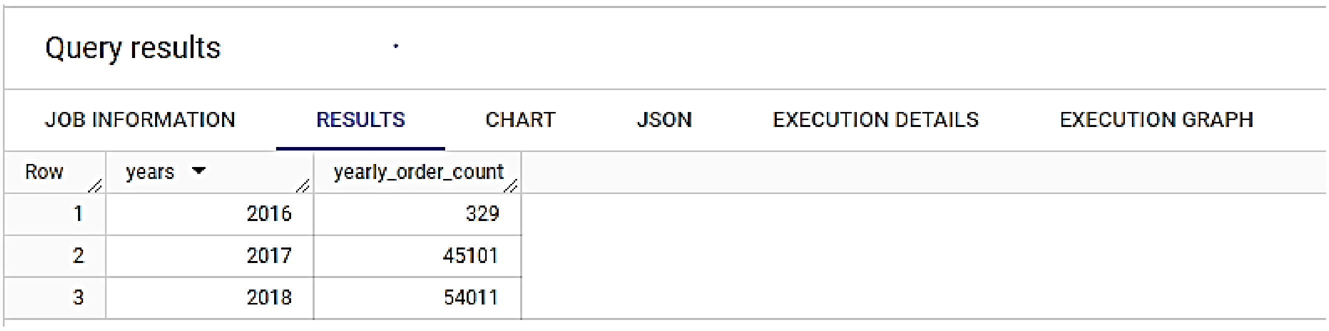
'TARGET' received orders from 4119 different cities and 27 different states.

2. In-depth Exploration: (Checking for trends in order or monthly seasonality)

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

QUERY:

```
select years, count(years) as yearly_order_count
from
    ( select
            extract(year from order_purchase_timestamp) as years,
    from BCS.orders ) as x
  group by years
  order by years
```



Row	years	yearly_order_count
1	2016	329
2	2017	45101
3	2018	54011

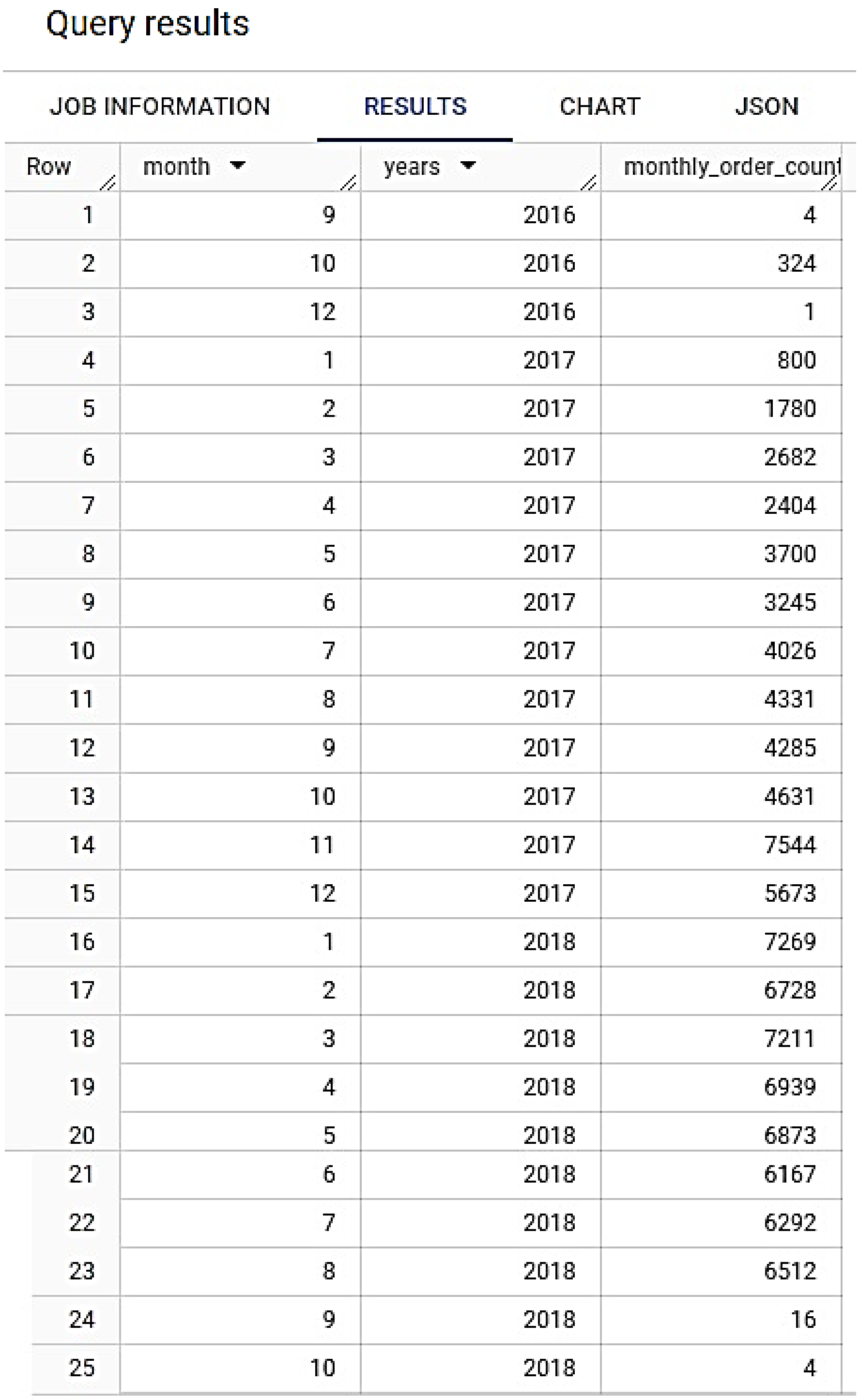
INFERENCE :

The number of Orders received by the TARGET increased over the past years. Initial response over few months was low but later the number of orders increased exponentially.

b. Checking for monthly seasonality in terms of the no. of orders being placed?

QUERY:

```
select  month, years,
        count(month) as monthly_order_count
from
    (select
            extract(year from order_purchase_timestamp) as years,
            extract(month from order_purchase_timestamp) as month
     from   BCS.orders ) as x
group by month, years
order by years, month
```


Query results

Row	month	years	monthly_order_count
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245
10	7	2017	4026
11	8	2017	4331
12	9	2017	4285
13	10	2017	4631
14	11	2017	7544
15	12	2017	5673
16	1	2018	7269
17	2	2018	6728
18	3	2018	7211
19	4	2018	6939
20	5	2018	6873
21	6	2018	6167
22	7	2018	6292
23	8	2018	6512
24	9	2018	16
25	10	2018	4

INFERENCE:

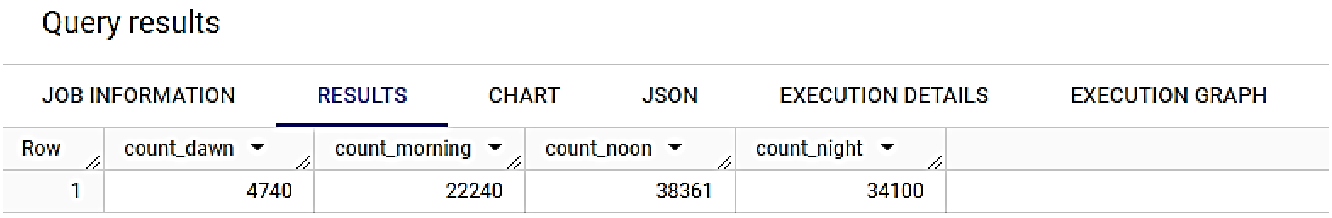
- ✓ **2016: The TARGET's data starts from September, which shows a noticeable increase in orders in October and then a drop in December.**
- ✓ **2017: whereas there is a consistent rise in the number of orders from January to November, peaking in November, followed by a decrease in December.**
- ✓ **2018: Orders start high in January, remain relatively high throughout the year, but there's a significant drop in September and October, followed by a small number of orders in the subsequent months.**

The query result highlights that there is no seasonality in the order data, but similarity particularly in the fall of number of orders can be noticed at the end of the years.

2 c. Checking for the Time of the day, during which the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

QUERY:

```
select
    count(case when order_time = 'dawn' then 1 end) as count_dawn,
    count(case when order_time = 'morning' then 1 end) as count_morning,
    count(case when order_time = 'noon' then 1 end) as count_noon,
    count(case when order_time = 'night' then 1 end) as count_night
from
    (select
        case
          when time >= '00:00:00' and time < '06:00:00' then 'dawn'
          when time >= '06:00:00' and time < '12:00:00' then 'morning'
          when time >= '12:00:00' and time < '18:00:00' then 'noon'
          else 'night'
        end as order_time
    from
        (select
          extract(time from order_purchase_timestamp) as time
        from bcs.orders ) as x ) as y
```



Row	count_dawn	count_morning	count_noon	count_night
1	4740	22240	38361	34100

INFERENCE:

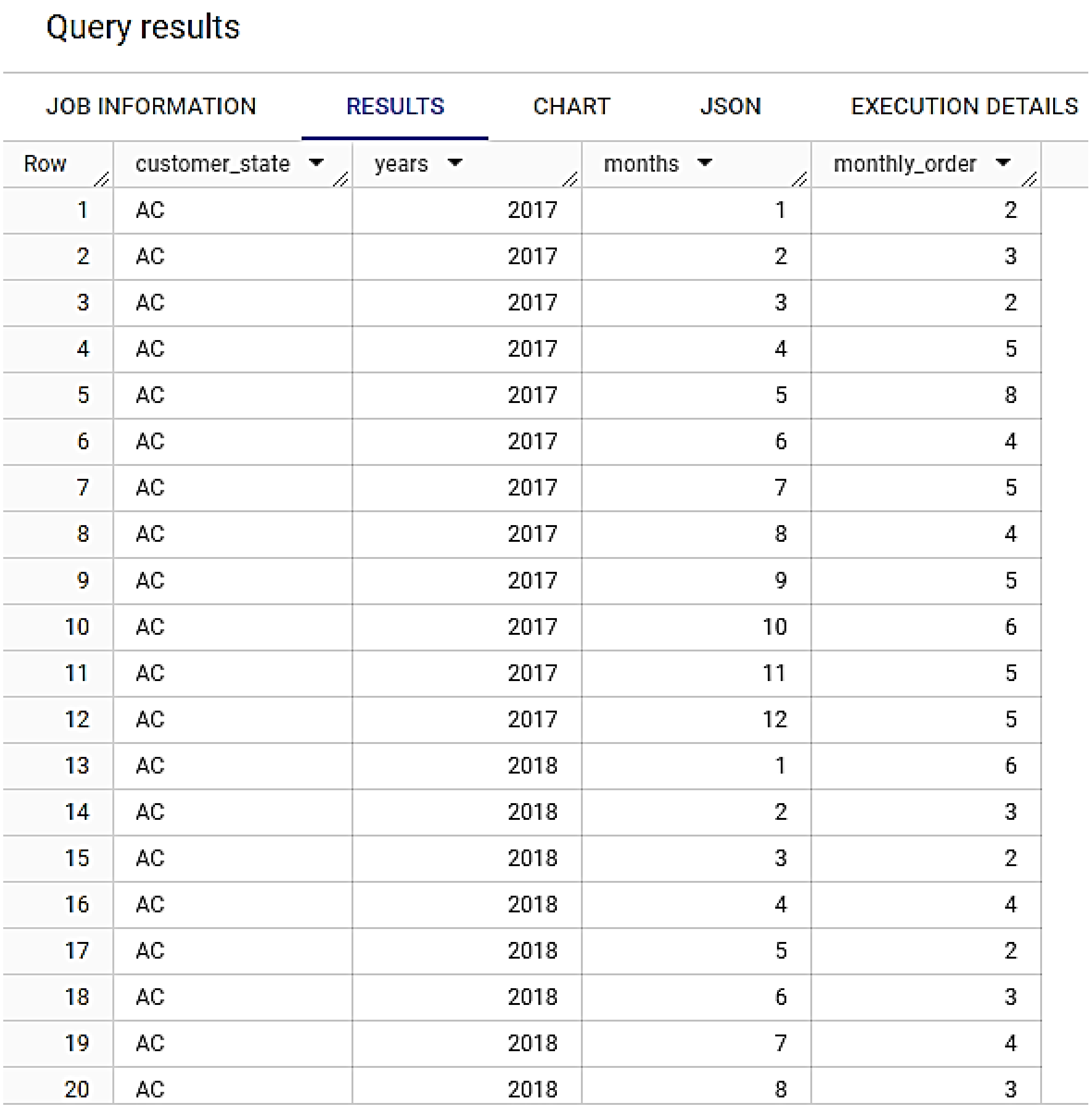
The result shows that people tend to order at TARGET more often in the noon and night than the morning or dawn time.

3. Checking for Geographical distribution of customers and trends

a. Month on Month no. of orders placed in each state.

QUERY:

```
select  customer_state,years, months,
        count(months) as monthly_order
from (
        select  c.customer_state,
                extract(month from o.order_purchase_timestamp) as months,
                extract(year from o.order_purchase_timestamp) as years
        from            BCS.orders o
        left join       BCS.customers c
        on              c.customer_id = o.customer_id
         ) as a
group by customer_state, years, months
order by customer_state, years, months
```



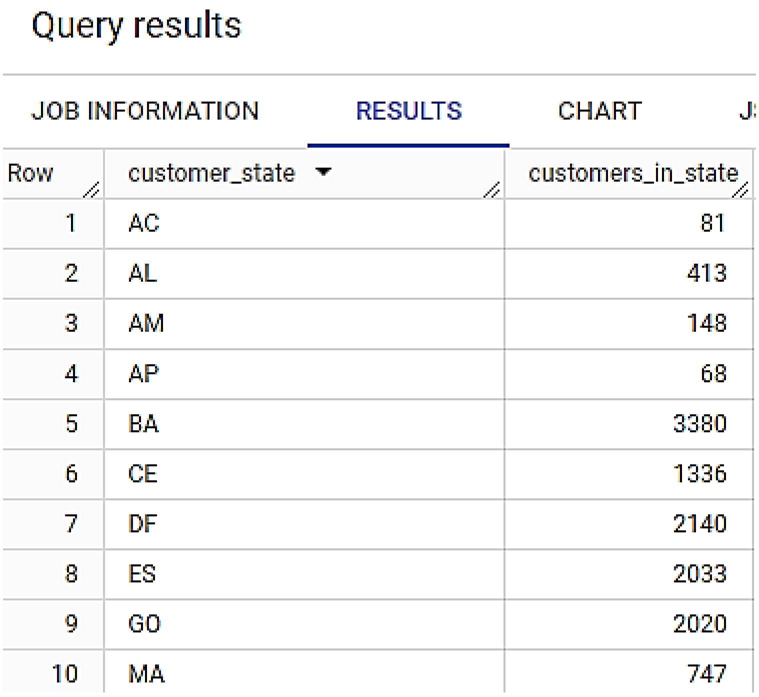
Row	customer_state	years	months	monthly_order
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3
15	AC	2018	3	2
16	AC	2018	4	4
17	AC	2018	5	2
18	AC	2018	6	3
19	AC	2018	7	4
20	AC	2018	8	3

INFERENCE:
The query output shows the number of orders received every month from each state.

3 b. Customers distribution across all the states?

QUERY:

```
select
    customer_state,
    count(customer_unique_id) as state_customers
from BCS.customers
group by customer_state
order by customer_state;
```



Query results

Row	customer_state	customers_in_state
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:

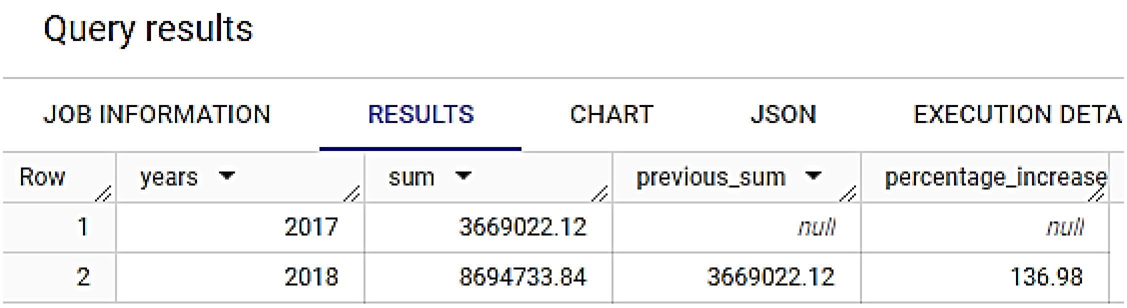
The query result provides the number of customers in every state.

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

a. Getting the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only). Using the "payment_value" column in the payments table to get the cost of orders.

QUERY:

```sql
with CTE1 as (
select
    o.order_id,
    p.payment_value,
    extract(year from o.order_purchase_timestamp) as years,
    extract(month from o.order_purchase_timestamp) as months
from BCS.payments p
inner join BCS.orders o
on o.order_id = p.order_id
order by years
),
CTE2 as
(
        select order_id, payment_value, years, months
        from CTE1
        where years > 2016 and months between 1 and 8
),
CTE3 as
(
select  years,
round(sum(payment_value),2) as sum
from CTE2
group by years
order by years
),
CTE4 as
(
        select  years, sum,
        lag(sum,1) over(order by years) as previous_sum
from    CTE3
order by years
)
select  years, sum, previous_sum,
        CASE
        WHEN previous_sum IS NULL THEN NULL
        ELSE round(((sum - previous_sum) / previous_sum) * 100,2)
        END AS percentage_increase_payment
    from CTE4
    order by years
```



Row	years	sum	previous_sum	percentage_increase
1	2017	3669022.12	*null*	*null*
2	2018	8694733.84	3669022.12	136.98

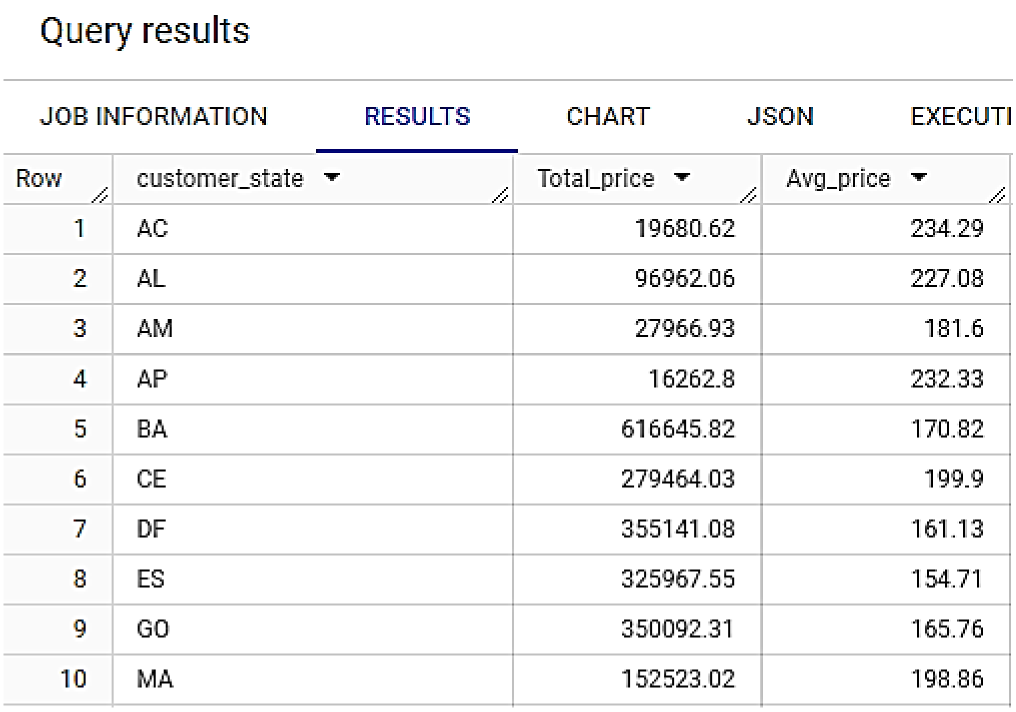
INFERENCE:

The payments received at TARGET between January and August in 2018 increased by 137% compared to the payments received during the same months in 2017.

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

QUERY:

```
select  customer_state,
        round(sum(payment_value),2) as Total_price,
        round(avg(payment_value),2) as Avg_price,
from (
        select  c.customer_state,o.order_id,
                p.payment_value
        from BCS.customers c
        join BCS.orders o
        on c.customer_id = o.customer_id
        right join      BCS.payments p
        on              p.order_id = o.order_id
        order by customer_state
      ) as a
group by customer_state
order by customer_state
```



Row	customer_state	Total_price	Avg_price
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

INFERENCE:

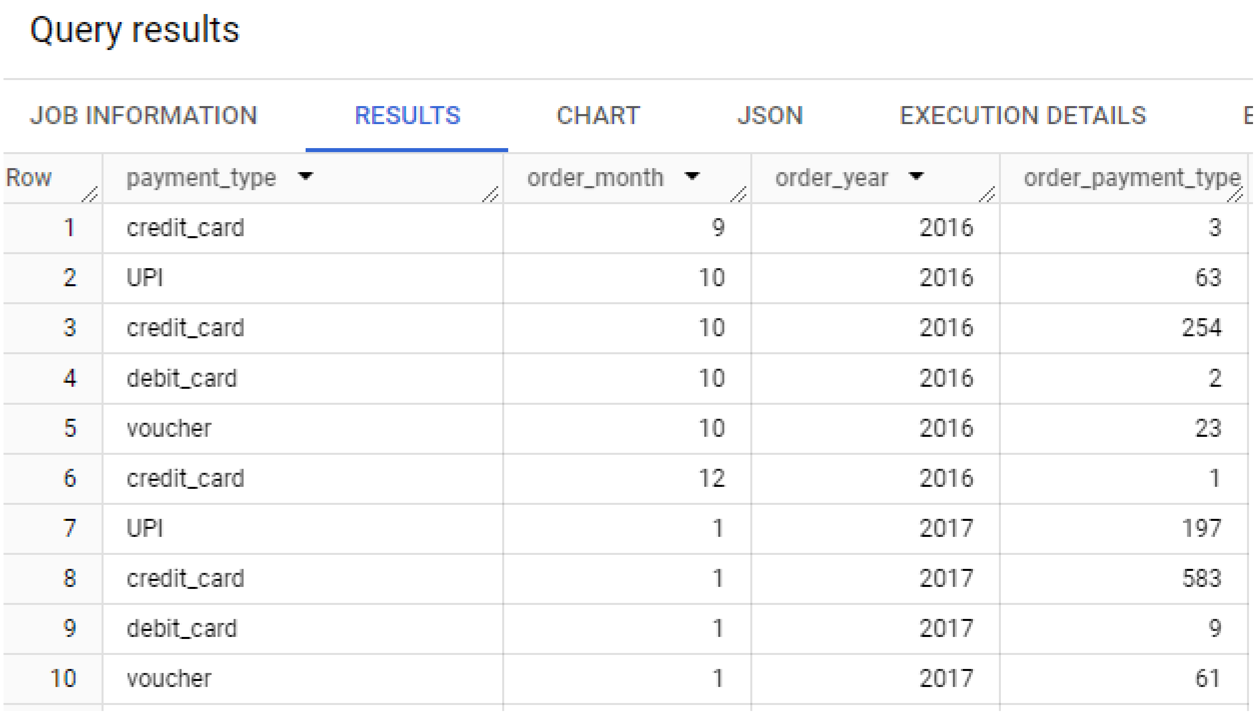
The query result provides the total price and average price of the total orders received per state at TARGET.

5. Analysis based on the payments:

a. Finding the month on month no. of orders placed using different payment types.

QUERY:

```
select  p.payment_type,
        extract(month from o.order_purchase_timestamp) as order_month,
        extract(year from o.order_purchase_timestamp) as order_year,
        count(*) as order_payment_type
from BCS.payments p
join BCS.orders o
on p.order_id = o.order_id
group by order_year, order_month, p.payment_type
order by order_year,  order_month, p.payment_type
```



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

INFERENCE:

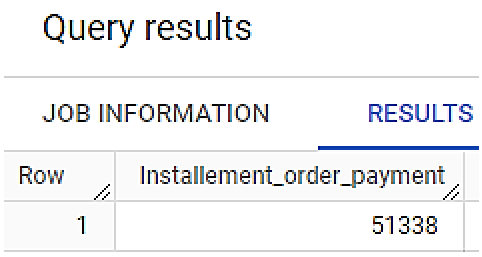
The data shows that there is a clear trend of increasing adoption and usage of digital payments, particularly credit cards and UPI, which dominate the transaction counts.

Based on the trends, it can be expected to have continued growth in digital payment transactions, with credit cards and UPI leading the way.

5 b. looking for the no. of orders placed on the basis of the payment instalments that have been paid.

QUERY:

```
select
        count(case when payment_installments > 1 then 1 end) as
Installement_order_payment
from BCS.payments p
```



Row	Installement_order_payment
1	51338

INFERENCE:

The query counts the number of payment transactions where the number of installments is greater than one.

The number of instalment based transactions (Installement_order_payment) indicates a significant customer preference for paying in installments rather than a single payment.

The popularity of installment payments can be leveraged in marketing campaigns to highlight the availability of flexible payment options, potentially attracting more customers.