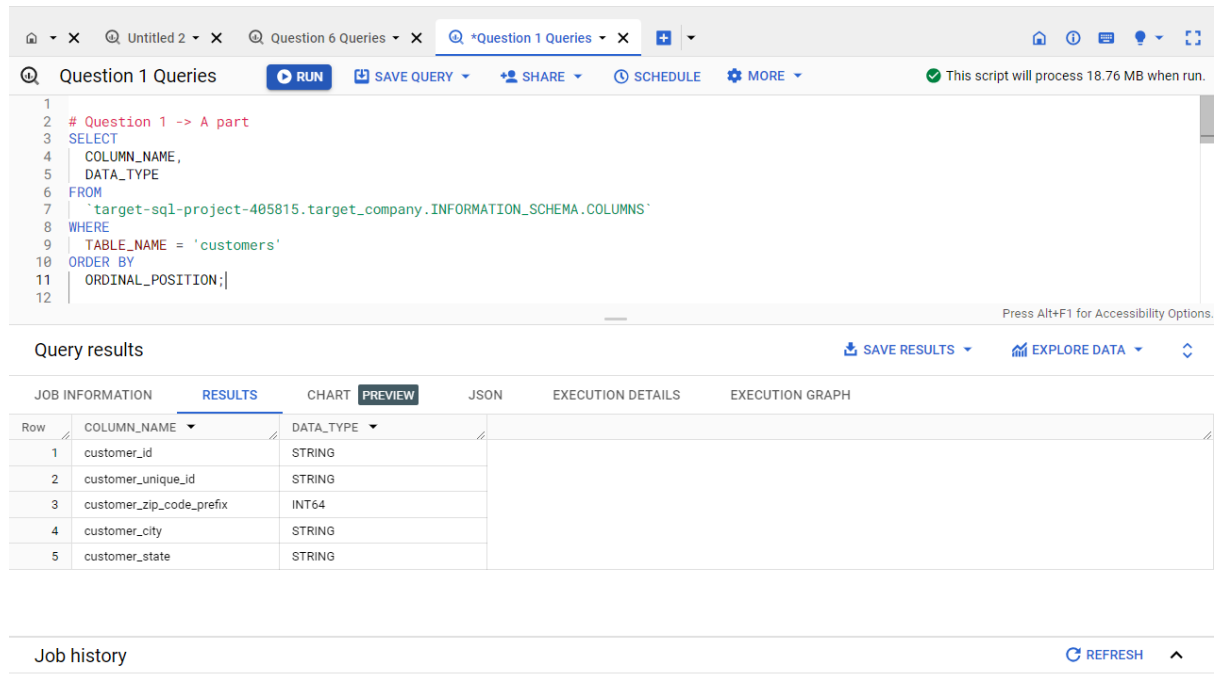


QUESTION 1 :

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

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



The screenshot displays a SQL query editor with the following query:

```
1
2 # Question 1 -> A part
3 SELECT
4     COLUMN_NAME,
5     DATA_TYPE
6 FROM
7     `target-sql-project-405815.target_company.INFORMATION_SCHEMA.COLUMNS`
8 WHERE
9     TABLE_NAME = 'customers'
10 ORDER BY
11     ORDINAL_POSITION;
12
```

Below the query editor, the "Query results" section is visible, showing a table with 5 columns:

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

Insight :-

In the customers table there are 5 columns where 4 columns are of String data type and one is Integer data type.

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

The screenshot shows a SQL query editor with the following code:

```
13  
14  
15 # Question 1 -> B part  
16  
17 select |  
18 min(order_purchase_timestamp) as `minimum_time`,  
19 max(order_purchase_timestamp) as `maximum_time`  
20 from `target_company.orders`;  
21  
22
```

Below the editor, the 'Query results' section is displayed. It includes tabs for 'JOB INFORMATION', 'RESULTS', 'CHART', 'PREVIEW', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'. The 'RESULTS' tab is active, showing a table with two columns: 'minimum_time' and 'maximum_time'. The table contains one row with the following values:

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

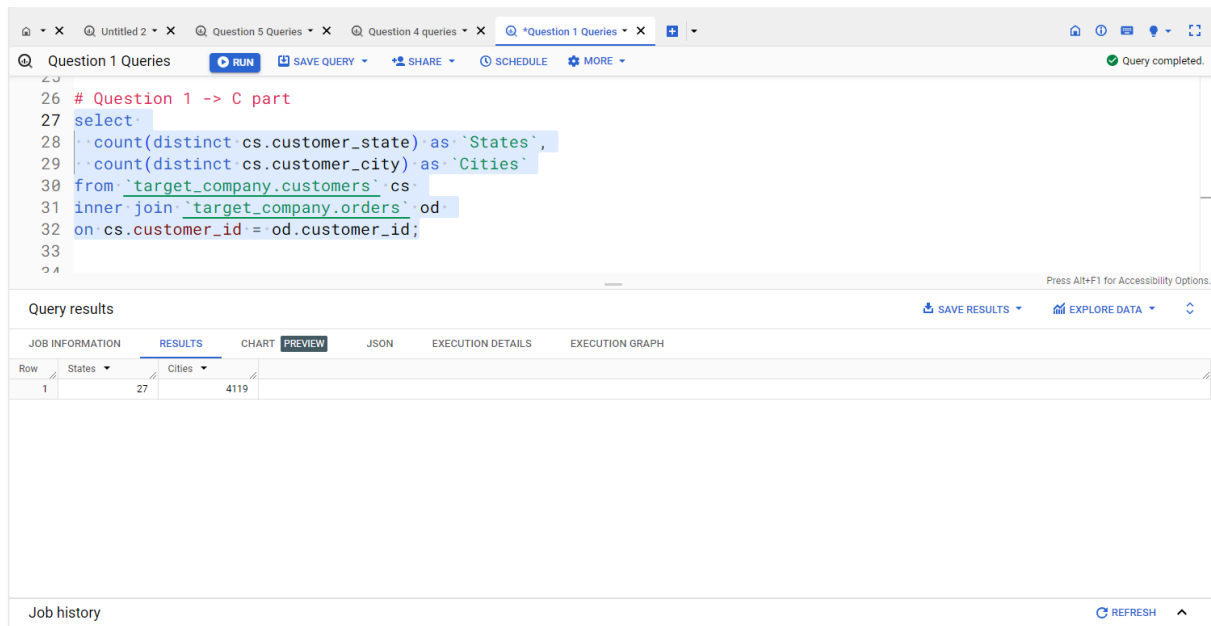
At the bottom of the interface, there is a 'Job history' section with a 'REFRESH' button.

Insight :-

The minimum time on which customers purchase the order is September 2016

The maximum time on which customers purchase the order is October 2018

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



The screenshot displays a SQL query editor with the following code:

```
26 # Question 1 -> C part
27 select
28     count(distinct cs.customer_state) as 'States',
29     count(distinct cs.customer_city) as 'Cities'
30 from 'target_company.customers' cs
31 inner join 'target_company.orders' od
32 on cs.customer_id = od.customer_id;
```

Below the query editor, the 'Query results' section is visible, showing a table with the following data:

Row	States	Cities
1	27	4119

The interface includes various controls such as 'RUN', 'SAVE QUERY', 'SHARE', 'SCHEDULE', and 'MORE' buttons. A 'Job history' section is also present at the bottom.

Insight :-

There are total 27 states and 4119 cities in the target dataset to extract the meaningful insights.

QUESTION 2 :

In-depth Exploration:

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

The screenshot shows a SQL query editor with the following query:

```
1 # Ques 2 part a
2 select
3   extract(year from order_purchase_timestamp) as 'year',
4   extract(month from order_purchase_timestamp) as 'month',
5   count(distinct order_id) as 'no_of_orders'
6 from 'target_company.orders'
7 group by 'year', 'month'
8 order by 'year', 'month';
```

Below the query editor, the 'Query results' section displays a table with the following data:

Row	year	month	no_of_orders
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

Insight :-

Yes! There is a growing trend in the no. of orders placed over the past years.

The orders increased in the year 2017 as compared to 2016

And in 2018 the orders placed are increased in only 2 months.

`no_of_orders` column shows a consistent increase over the years, it indicates a positive trend in the number of orders. This could be due to factors such as increased customer awareness, improved marketing strategies, or expanding product offerings.

Recommendation :-

Market Expansion: If the trend is positive, consider expanding your market reach. Explore new geographic areas or demographics to attract a wider customer base.

Customer Engagement: Focus on customer engagement and satisfaction. Collect feedback to identify areas for improvement and ensure a positive customer experience.

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

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

DISMISS UPGRADE

Untitled 2 Question 2 queries Question 1 Queries

Question 2 queries RUN SAVE QUERY SHARE SCHEDULE MORE Query completed.

```
15 # Ques 2 part b
16 select
17   FORMAT_DATETIME("%B", order_purchase_timestamp) AS `month`,
18   count(distinct order_id) as `no_of_sales`
19 from `target_company_orders`
20 where order_status = "delivered"
21 group by `month`
22 order by `no_of_sales` desc;
23
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

Job Information	Results	Chart	Preview	JSON	Execution Details	Execution Graph
Row	month	no_of_sales				
1	August	10544				
2	May	10295				
3	July	10031				
4	March	9549				
5	June	9234				
6	April	9101				
7	February	8208				
8	January	7819				
9	November	7289				
10	December	5514				

Results per page: 50 1 - 12 of 12

Insight :-

Yes we can see the monthly seasonality.

In the month of August there are maximum numbers of sales/orders are placed and in the month of September there are minimum numbers of sales/orders are placed.

Recommendation :-

Optimize inventory management based on seasonal demand. Ensure that you have sufficient stock during peak months and avoid overstocking during slower periods.

Align marketing and promotional activities with peak months to capitalize on increased customer engagement. Consider special offers or promotions during peak seasons.

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

Query

Question 2 queries

10

Ques 2 part c

with `day` as

(

select extract(hour from order_purchase_timestamp) as `hour`,

case

when

extract(hour from order_purchase_timestamp) between 0 and 6 then "Dawn"

when

extract(hour from order_purchase_timestamp) between 7 and 12 then "Morning"

when

extract(hour from order_purchase_timestamp) between 13 and 18 then "Afternoon"

else

"Night"

end as `time_of_the_day`,

count(distinct order_id) as `no_of_order`

from `target_company.orders`

group by `hour`, `time_of_the_day`

order by `hour`

)

select time_of_the_day, sum(no_of_order) as `total_orders_placed`

from day

group by time_of_the_day

order by `total_orders_placed` asc;

35

Query results

Result

Question 2 queries

10

Ques 2 part c

with `day` as

(

select extract(hour from order_purchase_timestamp) as `hour`,

case

when

extract(hour from order_purchase_timestamp) between 0 and 6 then "Dawn"

when

extract(hour from order_purchase_timestamp) between 7 and 12 then "Morning"

when

extract(hour from order_purchase_timestamp) between 13 and 18 then "Afternoon"

)

Query results

Row	time_of_the_day	total_orders_placed
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

Insight :-

The Brazilian customers purchases the most orders in the Afternoon then they purchases the most orders in the Night
Then they purchases the most orders in the Morning
And they purchases the least orders in the Dawn

Recommendation :-

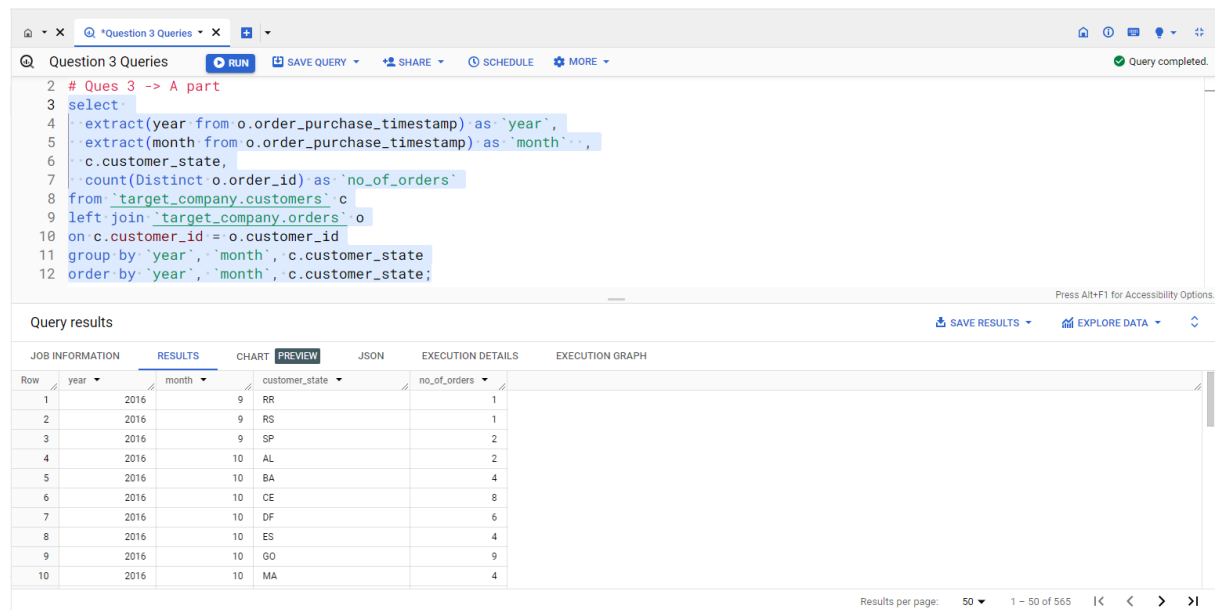
Create time-specific offers or discounts during periods with lower order activity to incentivize purchases during those times. This can help balance order distribution throughout the day.

If there are consistent peak times, ensure that your website or app provides a smooth and responsive user experience during these periods to accommodate increased traffic.

QUESTION 3 :

Evolution of E-commerce orders in the Brazil region:

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



```
2 # Ques 3 -> A part
3 select
4   extract(year from o.order_purchase_timestamp) as `year`,
5   extract(month from o.order_purchase_timestamp) as `month`,
6   c.customer_state,
7   count(Distinct o.order_id) as `no_of_orders`
8 from `target_company.customers` c
9 left join `target_company.orders` o
10 on c.customer_id = o.customer_id
11 group by `year`, `month`, c.customer_state
12 order by `year`, `month`, c.customer_state;
```

Query results

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	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

Insight :-

The number of highest orders placed in each month is SP state.

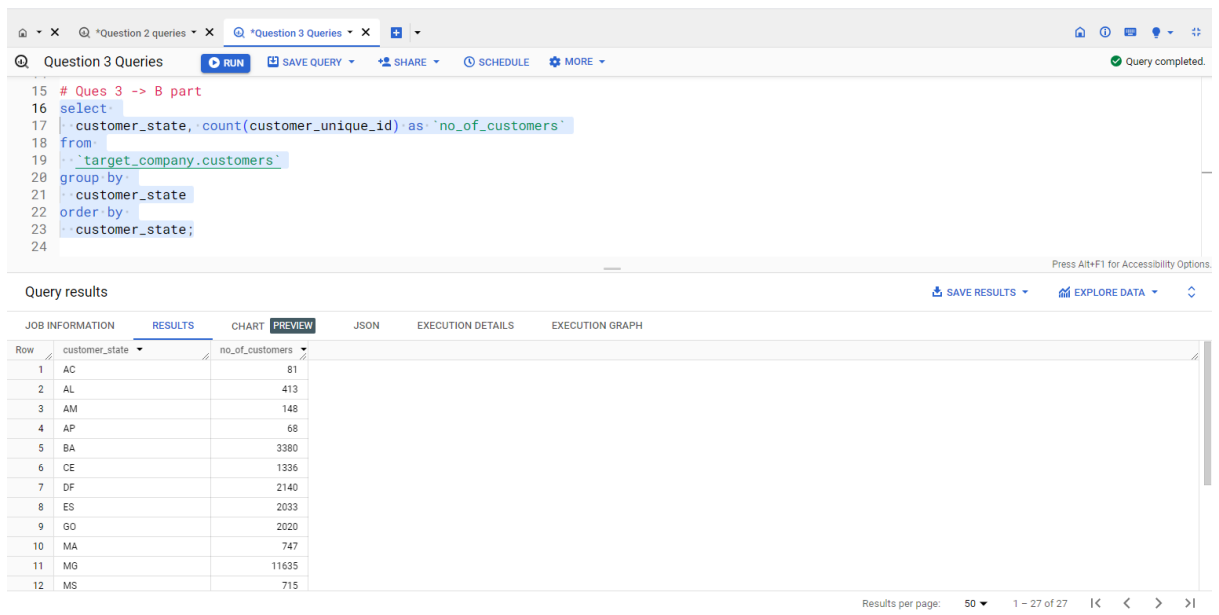
The number of lowest orders placed in each month is AP, RR state.

Recommendation :-

Optimize inventory levels in each state to meet demand. Adjust stock levels based on historical data to avoid stockouts during high-demand periods.

Ensure that customer support and logistics are well-equipped to handle variations in demand across different states. Provide adequate support during peak periods to maintain customer satisfaction.

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



The screenshot shows a SQL query editor with a query to count customers by state. Below the editor, the 'Query results' section displays a table with 12 rows of data. The table has two columns: 'customer_state' and 'no_of_customers'. The data is sorted by the number of customers in descending order.

```
15 # Ques 3 -> B part
16 select
17   customer_state, count(customer_unique_id) as 'no_of_customers'
18 from
19   `target_company.customers`
20 group by
21   customer_state
22 order by
23   customer_state;
24
```

Row	customer_state	no_of_customers
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
11	MG	11635
12	MS	715

Insight :-

The customers are distributed across different different states with the highest number of customers are in the state of SP (41746 customers)
The lowest number of customers are in the state of AP (68 customers).

Recommendation :-

Develop targeted marketing strategies for states with lower customer counts. This could include localized advertising, promotions, or other initiatives to increase brand awareness.

Explore opportunities for market expansion in states where there is untapped potential. Conduct market research to understand local preferences and adapt your offerings accordingly.

QUESTION 4 : Impact on 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 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

```
1 # Ques 4 part a
2
3 with `data_2017` as (
4 select
5   extract(year from o.order_purchase_timestamp) as `2017_Year`,
6   extract(month from o.order_purchase_timestamp) as `2017_Month`,
7   sum(p.payment_value) as `total_value_2017`
8 from `target_company.payments` p
9 left join `target_company.orders` o
10 on p.order_id = o.order_id
11 where
12   extract(year from o.order_purchase_timestamp) = 2017 and
13   extract(month from o.order_purchase_timestamp) between 1 and 8
14 group by `2017_Year`, `2017_Month`
15 order by `2017_Year`, `2017_Month`
16 ),
17
18 `data_2018` as (
19 select
20   extract(year from o.order_purchase_timestamp) as `2018_Year`,
21   extract(month from o.order_purchase_timestamp) as `2018_Month`,
22   sum(p.payment_value) as `total_value_2018`
23 from `target_company.payments` p
24 left join `target_company.orders` o
25 on p.order_id = o.order_id
26 where
27   extract(year from o.order_purchase_timestamp) = 2018 and
28   extract(month from o.order_purchase_timestamp) between 1 and 8
29 group by `2018_Year`, `2018_Month`
30 order by `2018_Year`, `2018_Month`
31 )
```

Result

Row	2017_Year	2017_Month	total_value_2017	orders_2017	2018_Year	2018_Month	total_value_2018	orders_2018	percentage_increase	percentage_increase_per_order
1	2017	1	138488.03999999999	850	2018	1	1115004.1800000001	7563	705.0	-9.51
2	2017	2	291908.00999999999	1886	2018	2	992463.3400000001	6952	240.0	-7.76
3	2017	3	449863.60000000004	2837	2018	3	1159652.1199999999	7512	158.0	-2.65
4	2017	4	417788.03000000004	2571	2018	4	1160785.4800000001	7209	178.0	-0.91
5	2017	5	592918.82000000004	3944	2018	5	1153982.1500000001	7135	95.0	7.58
6	2017	6	511276.38000000004	3436	2018	6	1023880.4999999999	6419	100.0	7.2
7	2017	7	592382.92000000004	4317	2018	7	1066540.7500000001	6507	80.0	19.45
8	2017	8	674396.32000000004	4550	2018	8	1022425.3199999999	6698	52.0	2.99

Insight :-

In the year of 2018 the orders of every month is increase as compared to the year of 2017 of every month.

In the year of 2018 the month between 3 and 5 the number of orders are maximum as compared to other months.

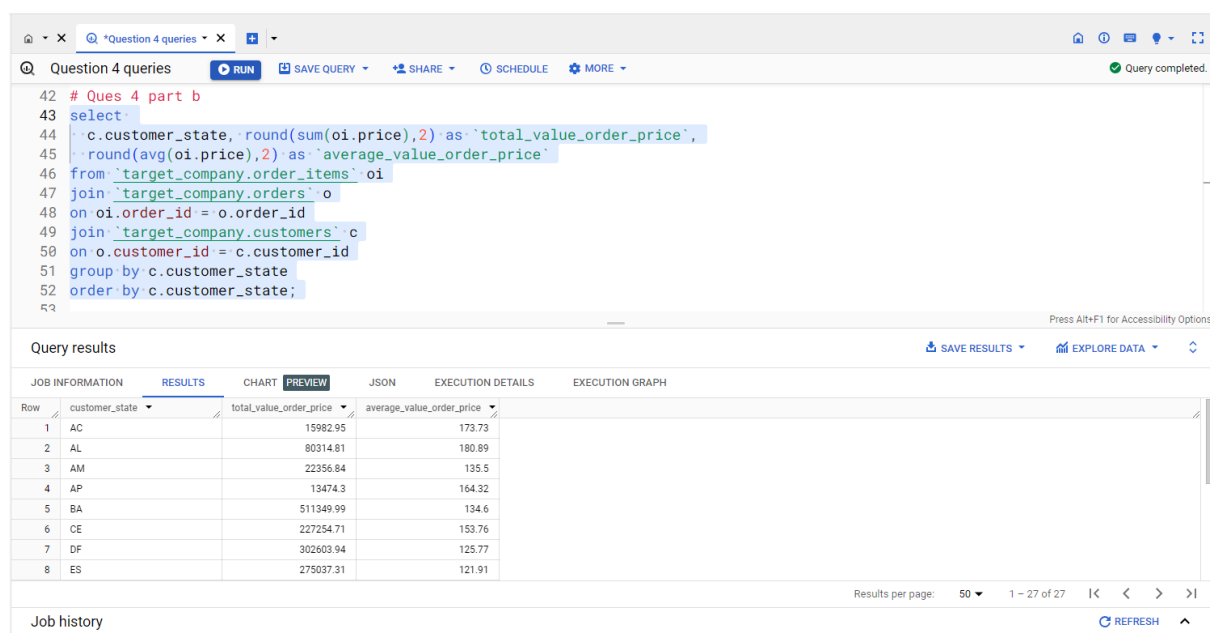
In the year 2018 the total value is increased in every month as compared to the year 2017.

Recommendation :-

Focus on enhancing the overall customer experience. Satisfied customers are more likely to place larger orders and become repeat customers.

Analyze customer feedback during the period of growth. Identify areas of improvement and implement changes based on customer suggestions.

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



Query results

Row	customer_state	total_value_order_price	average_value_order_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91

Insight :-

The maximum total value of order price in the state of SP with the amount of 5202955.05 and the minimum total value of order price in the state of RR with the amount of 7829.43.

The maximum average value of order price in the state of PB with the average amount of 216.67

And the minimum average value of order price in the state of SP with the average amount of 125.75.

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

The screenshot shows a SQL query editor with a query to calculate total and average freight values by state. Below the query, the 'Query results' section displays a table with 8 rows of data. The table has columns for 'customer_state', 'total_freight_value', and 'average_freight_value'. The results show that the state of AC has the highest total freight value (3686.75), while the state of RR has the lowest average freight value (17.27).

```
55 #Ques 4 part c
56 select
57   c.customer_state, round(sum(oi.freight_value),2) as 'total_freight_value',
58   round(avg(oi.freight_value),2) as 'average_freight_value'
59 from 'target_company.order_items' oi
60 join 'target_company.orders' o
61 on oi.order_id = o.order_id
62 join 'target_company.customers' c
63 on o.customer_id = c.customer_id
64 group by c.customer_state
65 order by c.customer_state;
```

Row	customer_state	total_freight_value	average_freight_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

Insight :-

The maximum total value of order freight price in the state of SP with the amount of 718723.07 and the minimum total value of order price in the state of RR with the amount of 2235.19.

The maximum average value of order price in the state of RR with the average amount of 48.59

And the minimum average value of order price in the state of SP with the average amount of 17.27.

QUESTION 5 : 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.

The screenshot shows the Google BigQuery interface. At the top, there's a 'Sandbox' banner. Below it, the query editor shows a SQL query for 'Question 5 part a'. The query selects distinct order IDs, calculates the time to delivery (difference between delivered and purchase dates), and calculates the difference between estimated and actual delivery dates. The query is executed, and the results are displayed in a table.

```
1 # Question 5 part a
2 select distinct order_id,
3    date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as `time_to_delivery`,
4    date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as `diff_estimated_delivery`
5 from `target_company.orders`
6 where (order_delivered_customer_date, order_purchase_timestamp) is not null and
7    order_status = "delivered"
8 order by order_id asc;
```

Query results

Row	order_id	time_to_delivery	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792cb16214	7	8
2	00018f77f2f0320c557190d7a144b0d3	16	2
3	000229ec398224ef6ca0657da4fc703e	7	13
4	00024acbcd0a6daa1e931b038114c75	6	5
5	00042b26cf59d7ce69dfab4e55b4f09	25	15
6	00048cc3ae777c65dbb7d2a0634bc1ea	6	14
7	00054e8431b9d7675808bcb819fb4a32	8	16
8	000576fe39319847cbb9d288c5617fa6	5	15
9	0005a1a1728c9d78508e2b08b904576c	9	0
10	0005f0442cb953dcd1d21e1fb923495	2	18

Results per page: 50 1 - 50 of 96478

Insight :-

These are the differences of estimated delivery time and actual delivery time .
In some case the orders are delivered before the estimated delivery time but in some cases the orders are delivered after the estimated delivery time.

Recommendation :-

If there are consistent delays, assess and optimize the delivery processes. Consider improvements in logistics, transportation, and order fulfillment to reduce delivery times.

Communicate estimated delivery dates transparently to customers. Manage expectations by providing accurate delivery estimates to minimize dissatisfaction due to delays.

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

(A) Top 5 states with Highest Average Freight Value

The screenshot shows a SQL query editor with the following code:

```
14
15
16 # Part 2 top 5 highest average freight value
17
18 select c.customer_state,
19        round(avg(oi.freight_value),2) as 'average_freight_value'
20 from 'target_company.order_items' oi
21 join 'target_company.orders' o
22 on oi.order_id = o.order_id
23 join 'target_company.customers' c
24 on o.customer_id = c.customer_id
25 group by c.customer_state
26 order by 'average_freight_value' desc
27 limit 5;
```

Below the query editor, the 'Query results' section displays a table with the following data:

Row	customer_state	average_freight_valu
1	RR	42.98
2	FB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

(B) Top 5 states with Lowest Average Freight Value

The screenshot shows a SQL query editor with the following code:

```
28
29 # Part 2 top 5 lowest average freight value
30
31 select c.customer_state,
32        round(avg(oi.freight_value),2) as 'average_freight_value'
33 from 'target_company.order_items' oi
34 join 'target_company.orders' o
35 on oi.order_id = o.order_id
36 join 'target_company.customers' c
37 on o.customer_id = c.customer_id
38 group by c.customer_state
39 order by 'average_freight_value' asc
40 limit 5;
```

Below the query editor, the 'Query results' section displays a table with the following data:

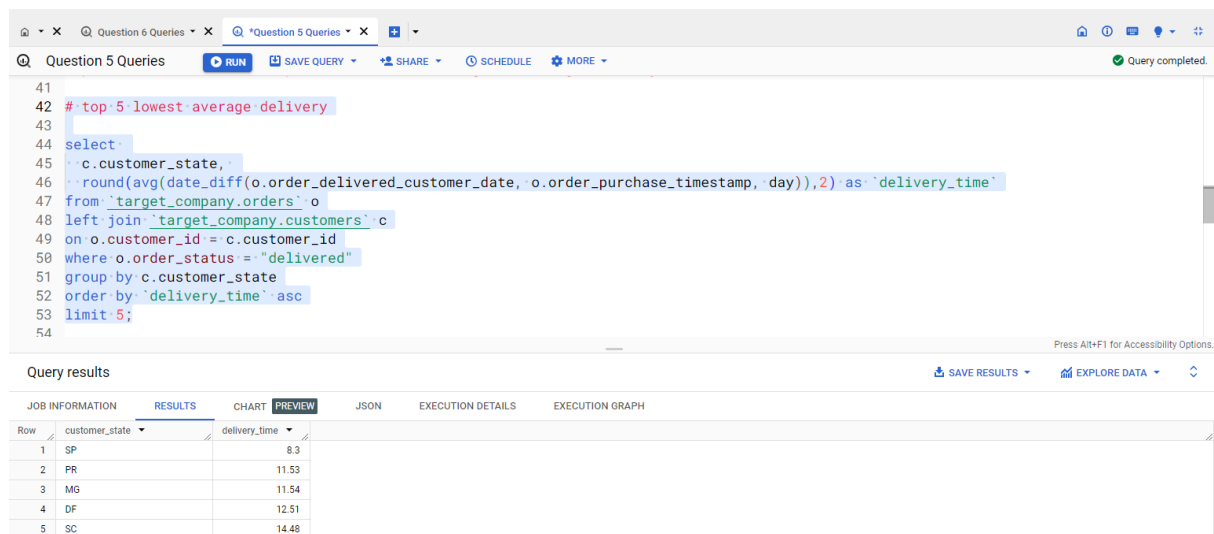
Row	customer_state	average_freight_valu
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

Insight :-

The details are extracted to get the top 5 states for highest and lowest average freight value.

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

Top 5 states with lowest average delivery time

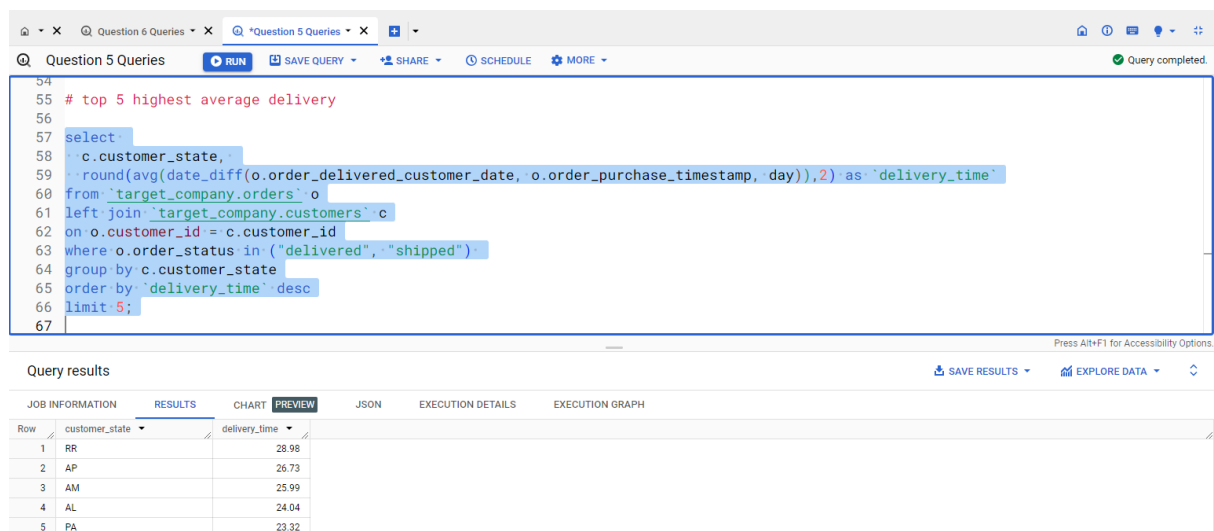


```
41
42 # top 5 lowest average delivery
43
44 select
45   c.customer_state,
46   round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, 'day')), 2) as 'delivery_time'
47 from 'target_company.orders' o
48 left join 'target_company.customers' c
49 on o.customer_id = c.customer_id
50 where o.order_status = "delivered"
51 group by c.customer_state
52 order by 'delivery_time' asc
53 limit 5;
```

Query results

Row	customer_state	delivery_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

Top 5 states with highest average delivery time



```
54
55 # top 5 highest average delivery
56
57 select
58   c.customer_state,
59   round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, 'day')), 2) as 'delivery_time'
60 from 'target_company.orders' o
61 left join 'target_company.customers' c
62 on o.customer_id = c.customer_id
63 where o.order_status in ("delivered", "shipped")
64 group by c.customer_state
65 order by 'delivery_time' desc
66 limit 5;
```

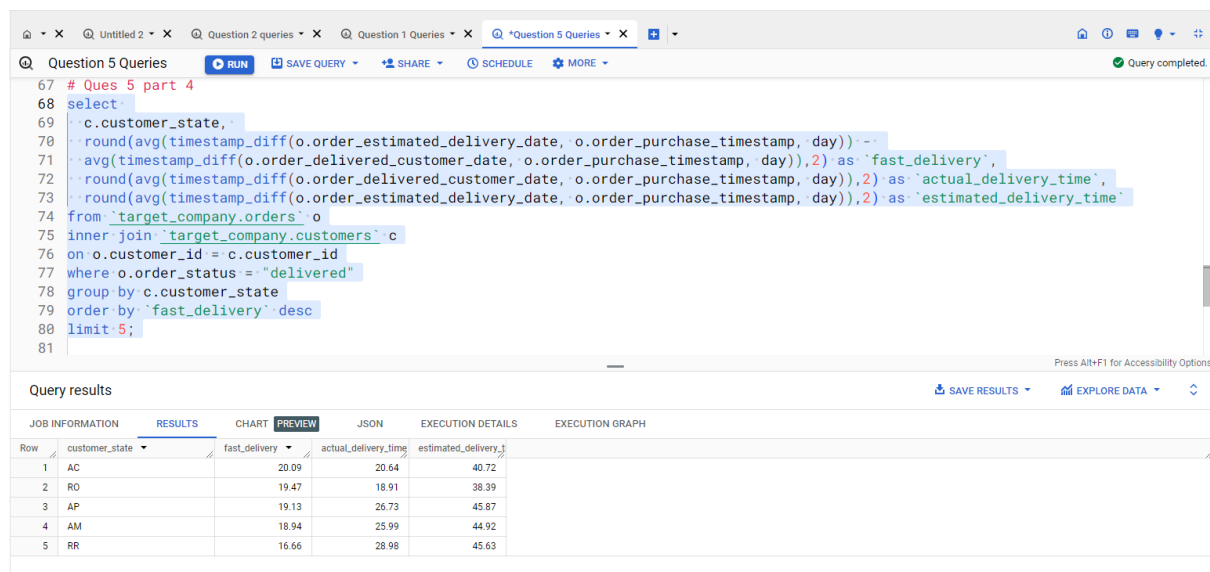
Query results

Row	customer_state	delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

Insight :-

The details are extracted to get the top 5 states with highest and 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.



The screenshot shows a SQL query editor with a query for finding the top 5 states with the fastest delivery times. The query calculates the difference between the actual delivery time and the estimated delivery time, rounded to two decimal places. The results table below shows the top 5 states: AC, RO, AP, AM, and RR, with their respective fast_delivery, actual_delivery_time, and estimated_delivery_time values.

```
67 # Ques 5 part 4
68 select
69   c.customer_state,
70   round(avg(timestamp_diff(o.order_estimated_delivery_date, o.order_purchase_timestamp, day))) --
71   avg(timestamp_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)), 2) as 'fast_delivery',
72   round(avg(timestamp_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)), 2) as 'actual_delivery_time',
73   round(avg(timestamp_diff(o.order_estimated_delivery_date, o.order_purchase_timestamp, day)), 2) as 'estimated_delivery_time'
74 from 'target_company.orders' o
75 inner join 'target_company.customers' c
76 on o.customer_id = c.customer_id
77 where o.order_status = "delivered"
78 group by c.customer_state
79 order by 'fast_delivery' desc
80 limit 5;
81
```

Row	customer_state	fast_delivery	actual_delivery_time	estimated_delivery_t
1	AC	20.09	20.64	40.72
2	RO	19.47	18.91	38.39
3	AP	19.13	26.73	45.87
4	AM	18.94	25.99	44.92
5	RR	16.66	28.98	45.63

Insight :-

These are the top 5 states where the delivery is really fast as compared to the estimated date of delivery .

The states are

- AC
- RO
- AP
- AM
- RR

QUESTION 6 : Analysis based on the payments:

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

```
# Q6 -> A part
select
  extract(year from o.order_purchase_timestamp) as 'year',
  extract(month from o.order_purchase_timestamp) as 'month',
  count(distinct p.order_id) as 'no_of_orders', p.payment_type
from target_company.payments p
join target_company.orders o
on p.order_id = o.order_id
group by 'year', 'month', p.payment_type
order by 'year', 'month', p.payment_type;
```

Query results					Press Alt+F1 for Accessibility Options.	
JOB INFORMATION					SAVE RESULTS	EXPLORE DATA
RESULTS						
CHART PREVIEW						
JSON						
EXECUTION DETAILS						
EXECUTION GRAPH						
Row	year	month	no_of_orders	payment_type		
1	2016	9	3	credit_card		
2	2016	10	63	UPI		
3	2016	10	253	credit_card		
4	2016	10	2	debit_card		
5	2016	10	11	voucher		
6	2016	12	1	credit_card		
7	2017	1	197	UPI		
8	2017	1	582	credit_card		
9	2017	1	9	debit_card		
10	2017	1	33	voucher		
11	2017	1	33	UPI		

Results per page: 50 1 - 50 of 90

Insight :-

The query provides a breakdown of the number of orders placed each month, categorized by payment type.

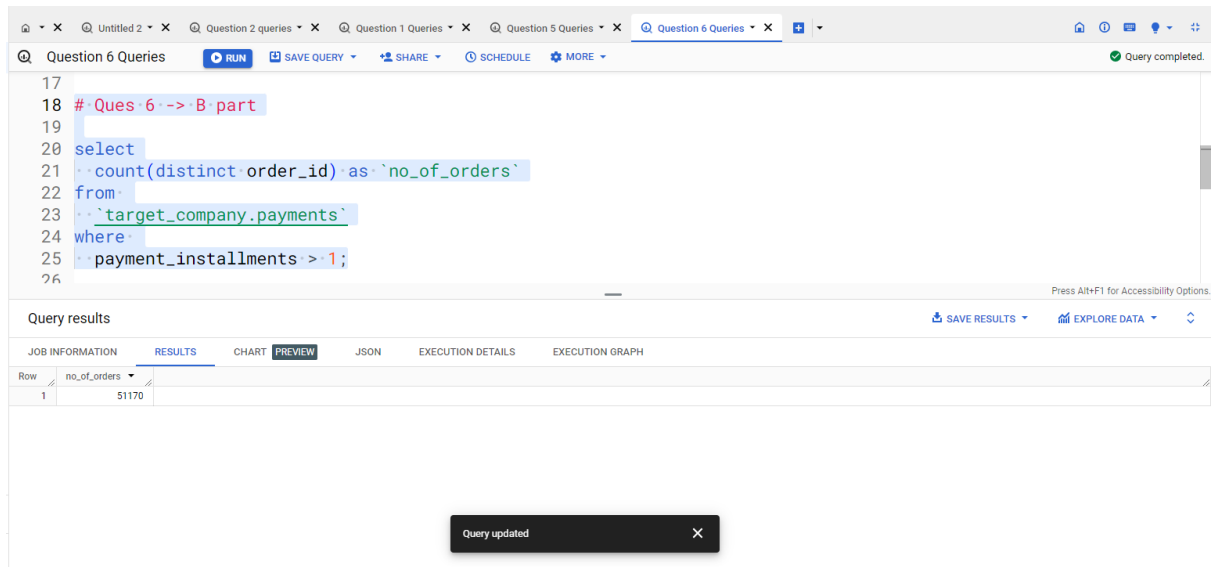
There are 5 types of payment

- Credit Card
- Debit Card
- UPI
- Voucher
- Not Defined

The most used payment type are Credit Card, UPI and Voucher.

The least used payment type are Debit Card and Not Defined.

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



The screenshot shows a SQL query editor interface. The query is as follows:

```
17  
18 # Ques 6 -> B part  
19  
20 select  
21 count(distinct order_id) as `no_of_orders`  
22 from  
23 `target_company.payments`  
24 where  
25 payment_installments > 1;  
26
```

Below the query editor, the 'Query results' section is visible, showing a table with one row and one column:

Row	no_of_orders
1	51170

A 'Query updated' notification is displayed at the bottom of the interface.

Insight :-

The no. of orders placed on the basis of payment installments is 51170.