

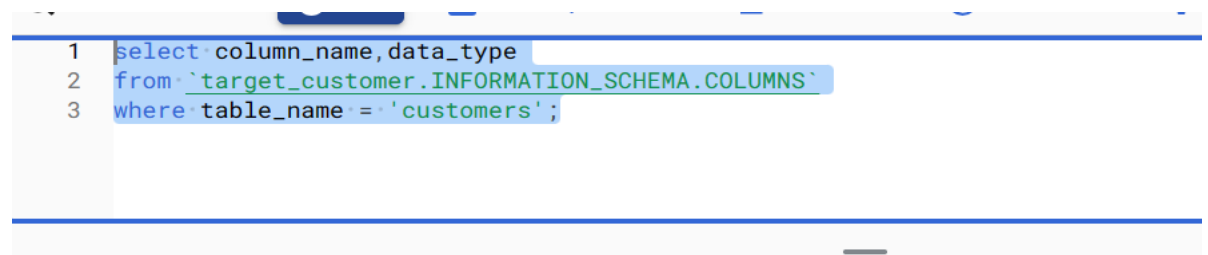
Target Business case study – submitted by Omprakash S

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

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

1.1) Answer – Query

```
select column_name,data_type
from `target_customer.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers';
```



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

```
1 select column_name,data_type
2 from `target_customer.INFORMATION_SCHEMA.COLUMNS`
3 where table_name = 'customers';
```

Below the query editor, the "Query results" section is displayed. It includes a "SAVE RESULTS" button and a table with the following data:

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS
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			

Insights : I observe the data type of the customer's table. By the same way we can find the data type of the necessary tables and this will help us to find the relationship between the tables and get the ER diagram of the entire data sets and to know the data structure.

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

1.2) Answer **Query :**

```
select min(order_purchase_timestamp) as start_date,
max(order_purchase_timestamp) as end_date
from `target_customer.orders`
```

Result :

The screenshot shows a SQL query editor interface. On the left is an 'Explorer' panel with a search bar and a list of resources: geolocation, order_items, order_reviews, orders, payments, products, and sellers. The main editor area has a tab titled 'Untitled' with a 'RUN' button. The query text is:

```
1 select min(order_purchase_timestamp) as start_date,
2 max(order_purchase_timestamp) as end_date
3 from `target_customer.orders`
4
5
```

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

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

Insight : I found that the sales of orders started at 4th Sep 2016 and ended at 17th oct 2018. This helps to know the customers order frequency between the time period.

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

1.3) Answer

Query :

```
select count(distinct customer_city) as count_city, count(distinct
customer_state) as count_state
from `target_customer.customers`
```

Result :

The screenshot shows a SQL query editor interface. The main editor area has a tab titled 'Untitled' with a 'RUN' button. The query text is:

```
1 select count(distinct customer_city) as count_city, count(distinct customer_state) as count_state
2 from `target_customer.customers`
3
4
5
```

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

Row	count_city	count_state
1	4119	27

Insight : It is observed that customers are from 4119 cities and from 27 different states of Brazil.

Recommendation : This data helps us to focus on the deep into city from which more orders are received, so we can stock the more products near to the location. This will help in to focus the cost to company for warehouses, staff workers and delivery partners in the near by cities and states.

2.In-depth Exploration:

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

2.1) Answer

Query : `select pre_year, count(order_id) as count_of_orders
from
(select *, extract (year from order_purchase_timestamp) as pre_year
from `target_customer.orders`) tb
group by pre_year`

Result :

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Untitled', 'orders', and 'Untitled 2'. The 'Untitled 2' tab is active, showing a SQL query. Below the query editor, there are buttons for 'RUN', 'SAVE', 'DOWNLOAD', and 'SHARE'. The query results are displayed in a table with the title 'Query results'. The table has columns for 'Row', 'pre_year', and 'count_of_orders'. The results show three rows of data for the years 2016, 2017, and 2018.

Row	pre_year	count_of_orders
1	2016	329
2	2017	45101
3	2018	54011

Insights :

1. We can understand that there is a steep increase between the year 2016 – 2017 with 44,772 no of orders. But between the year 2017-2018, there is a decrease in no of orders with only 8,910 no of orders when compared to previous year

2. This provides us with the number of orders by the customers.

Recommendations :

To increase the count, we need to focus more on the customer needs from his end like focusing on the products image with high resolution or a small video of the products, User interface design, enable more filter options, enable best customer support.

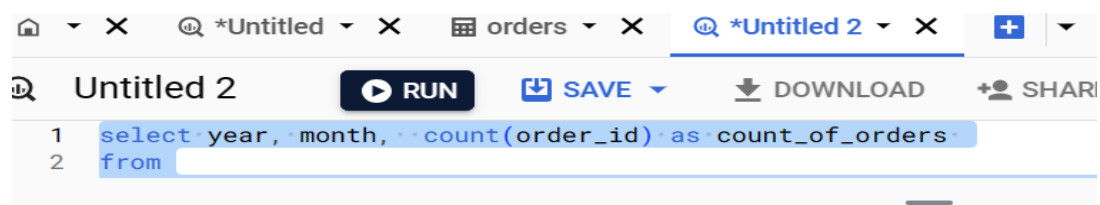
2.2) Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

2.2) Answer

Query :

```
select year, month, count(order_id) as count_of_orders
from
(select *, extract (year from order_purchase_timestamp) as
year,extract(month from order_purchase_timestamp ) as month
from `target_customer.orders` ) tb
group by year,month
order by year,month
```

Result :



Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION
Row	year	month	count_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		

Insights: This can help to find the minimum order placed and maximum orders placed in a month and also the increase in the no of orders placed by the customers, also how many new customers are joined the e-commerce to order products.

Recommendations: Create awareness programmes like advertisements, local sponsorship, digital ads for new customer to login and purchase products.

Providing discount for first purchase for new customers and on board a greater number of vendors to sell their products on company's e-commerce website.

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

2.3) Answer

Query :

```
select time_of_order,count(*) as no_of_orders
from(
select
case
when time_ between 0 and 6 then 'Dawn'
when time_ between 7 and 12 then 'Mornings'
when time_ between 13 and 18 then 'Afternoon'
when time_ between 19 and 23 then 'Night'
end as time_of_order,time_
from (
select extract(hour from order_purchase_timestamp) as time_
from `target_customer.orders` ) A ) B
group by time_of_order
```

```
87 select time_of_order,count(*) as no_of_orders
88 from(
89 select
90 case
91 when time_ between 0 and 6 then 'Dawn'
92 when time_ between 7 and 12 then 'Mornings'
93 when time_ between 13 and 18 then 'Afternoon'
94 when time_ between 19 and 23 then 'Night'
95 end as time_of_order,time_
96 from (
97 select extract(hour from order_purchase_timestamp) as time_
98 from `target_customer.orders` ) A ) B
99 group by time_of_order
```

Query results

[SAVE RI](#)

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DET
row	time_of_order	no_of_orders			
1	Mornings	27733			
2	Dawn	5242			
3	Afternoon	38135			
4	Night	28331			

Insights: The result shows that the people of Brazil purchase during afternoon followed by Night, Mornings and Dawn. So, it helps us to focus on availability, scalability of the website and to access it with out any interruptions and down time. Most of the customers are free during afternoon and night so delivery of products should be around this time period.

Recommendations: To increase the sales, offer can be announced during afternoon to mid night.

SMS related messages can be sent to customer during this time period for any announcements.

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

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

3.1) Answer Query :

```
select tb.customer_state,tb.month, count(tb.order_id) as
Number_of_orders
from(
select *,extract(month from order_purchase_timestamp) as month
from `target_customer.orders` `o` join `target_customer.customers` `c`
on o.Customer_id = c.customer_id
join `target_customer.geolocation` `g`
on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
) tb
group by tb.customer_state,tb.month
order by tb.customer_state,tb.month
```

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTI
Row	customer_state	month	Number_of_orders			
1	AC	1	692			
2	AC	2	512			
3	AC	3	514			
4	AC	4	785			
5	AC	5	1154			

Results per page: 50 1 – 50 of 322

Insights : For each state, we can observe the number of orders placed by customer on monthly basis. With min aggregate function I found that only 49 orders received from AM state during October month. With max, it is found that 660765 orders received from SP state during August. This helps us to find the customer purchasing behaviour and how many new customers have ordered on monthly basis.

Recommendation : we can provide a standard additional discount for first order so it will help to encourage people to buy products from e-commerce.

We can focus on logistics, delivery partners in the state with high orders like hiring delivery partner for fast delivery.

We can find the most ordered products so it will help us to restock the products in the location.

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

3.2) Answer
Query

```
select g.geolocation_state,count(c.customer_id) as no_of_customers
      from `target_customer.geolocation` g join
      `target_customer.customers` c
      on g.geolocation_zip_code_prefix =
      c.customer_zip_code_prefix
      group by g.geolocation_state
      order by no_of_customers desc
```

target project ...

▶ RUN

SAVE QUERY

▼

+

SHARE

▼

🕒 SCHEDULE

1

select g.geolocation_state, count(c.customer_id) as no_of_customers

from `target_customer.geolocation` g join `target_customer.customers` c

on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix

group by g.geolocation_state

order by no_of_customers desc

7

Press Alt

Query results

SAVE RESULTS

▼

EX

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTIC

Row	geolocation_state	no_of_customers
1	SP	5620430
2	RJ	3015690
3	MG	2878728
4	RS	805370
5	PR	626021

Results per page:

50

▼

1 - 27 of 27

Insights : The highest number of customers are from the state SP with fifty-six lakh twenty thousand four thirty. The lowest number of customers are from the state RR with two thousand eight seven.

Recommendations: For the state holding highest number of customers, we need to focus on customer retention liking better delivery experience, increase customer support to resolve issues as quickly, creating necessary infrastructure facility in the ware-house in the state.

For the state with lowest number of customers, the marketing and branding department has to take concern step to reach the customer. Getting feedback of existing customer to understand user experience and improving on it. Organise campaigns to create awareness about the company and sponsor the local events in crowded areas like malls, markets areas provide them with coupons for their first purchases. This might help to increase the customer base in the states having a smaller number of the customers.

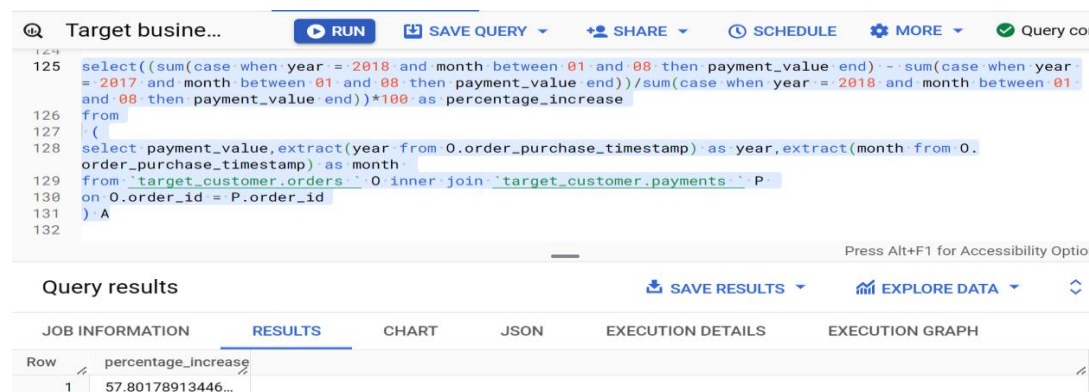
4.Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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.

4.1) answer

```
Query: select((sum(case when year = 2018 and month between 01 and 08 then payment_value end) - sum(case when year = 2017 and month between 01 and 08 then payment_value end))/sum(case when year = 2018 and month between 01 and 08 then payment_value end))*100 as percentage_increase
from
(
select payment_value,extract(year from O.order_purchase_timestamp) as year,extract(month from O.order_purchase_timestamp) as month
from `target_customer.orders` O inner join `target_customer.payments` P
on O.order_id = P.order_id
) A
```

Result :



Target busine... RUN SAVE QUERY SHARE SCHEDULE MORE Query coi

```
125 select((sum(case when year = 2018 and month between 01 and 08 then payment_value end) - sum(case when year = 2017 and month between 01 and 08 then payment_value end))/sum(case when year = 2018 and month between 01 and 08 then payment_value end))*100 as percentage_increase
126 from
127 (
128 select payment_value,extract(year from O.order_purchase_timestamp) as year,extract(month from O.order_purchase_timestamp) as month
129 from `target_customer.orders` O inner join `target_customer.payments` P
130 on O.order_id = P.order_id
131 ) A
132
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	percentage_increase				
1	57.80178913446...				

Insights : There is a 58-percentage increase in cost of orders which is positive development. This shows the order value placed by the customers is high in 2018 when compared to 2017(Jan to Aug)

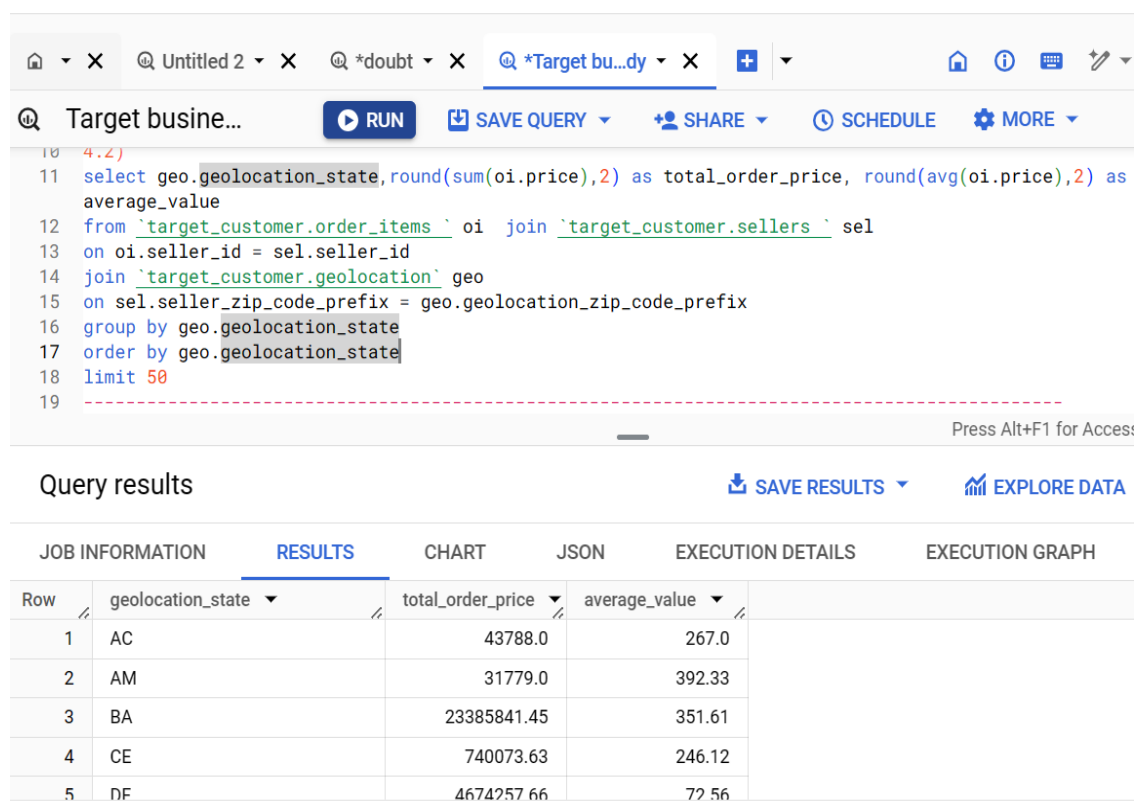
Recommendation: We can further investigate the frequent orders placed by the customer with high value so to increase the availability of the products from vendors.

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

4.2) answer

Query :

```
select geo.geolocation_state,sum(oi.price) as total_order_price,
avg(oi.price) as average_value
from `target_customer.order_items` oi join `target_customer.sellers` sel
on oi.seller_id = sel.seller_id
join `target_customer.geolocation` geo
on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
group by geo.geolocation_state
order by geo.geolocation_state
limit 50
```



The screenshot shows a SQL query editor with a query that calculates the total and average order price for each state. The query is executed, and the results are displayed in a table. The table has columns for Row, geolocation_state, total_order_price, and average_value. The results show data for states AC, AM, BA, CE, and DF.

Row	geolocation_state	total_order_price	average_value
1	AC	43788.0	267.0
2	AM	31779.0	392.33
3	BA	23385841.45	351.61
4	CE	740073.63	246.12
5	DF	4674257.66	72.56

Insights : I could observe that total order price of each state gives us the information about the total amount spent by customer on products by this we can get customer spending cost by each state where it will help to segregate us the high-capacity spending customers by each state. The average value column tells that at least spending of customers for any order.

Recommendation : Based on the insights, we can detect the major sales in particular state. This helps us get the revenue information.

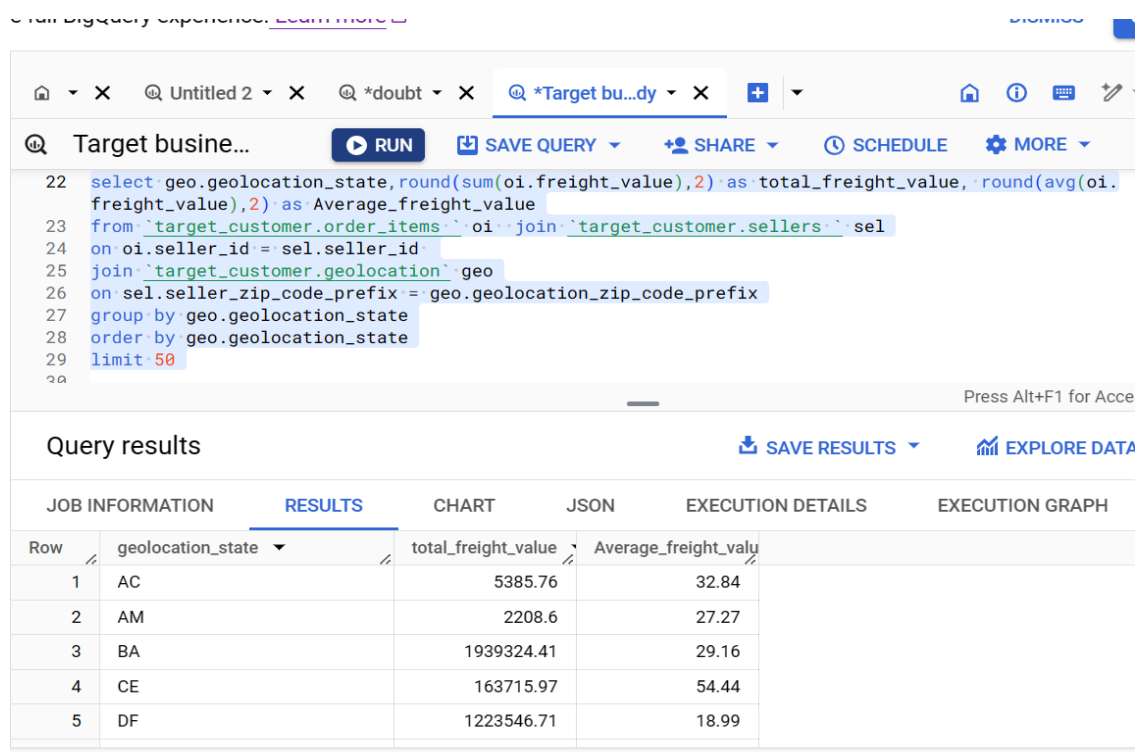
3) Calculate the Total & Average value of order freight for each state.

4.3) answer

Query

```
select geo.geolocation_state, round(sum(oi.freight_value),2) as
total_freight_value, round(avg(oi.freight_value),2) as Average_freight_value
from `target_customer.order_items` oi join `target_customer.sellers` sel
on oi.seller_id = sel.seller_id
join `target_customer.geolocation` geo
on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
group by geo.geolocation_state
order by geo.geolocation_state
limit 50
```

Run BigQuery query experience. [Learn more](#)



Target busine... RUN SAVE QUERY SHARE SCHEDULE MORE

```
22 select geo.geolocation_state, round(sum(oi.freight_value),2) as total_freight_value, round(avg(oi.
23 freight_value),2) as Average_freight_value
24 from `target_customer.order_items` oi join `target_customer.sellers` sel
25 on oi.seller_id = sel.seller_id
26 join `target_customer.geolocation` geo
27 on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
28 group by geo.geolocation_state
29 order by geo.geolocation_state
30 limit 50
```

Press Alt+F1 for Acc...

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	geolocation_state	total_freight_value	Average_freight_valu			
1	AC	5385.76	32.84			
2	AM	2208.6	27.27			
3	BA	1939324.41	29.16			
4	CE	163715.97	54.44			
5	DF	1223546.71	18.99			

Insights : The total freight value cost indicates the total delivery cost for each which basically means that one among the expenditure cost to sell the products and average freight value gives us the at least amount incurred in the logistics

Recommendations: We can create a limit to no of orders placed eligible for free delivery to customers.

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.

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_delivered_customer_date - order_estimated_delivery_date

5.1)

Answer

Query

```
select
date_diff(date(order_delivered_customer_date),date(order_purchase_timestamp)
,day) as dilever_time,
date_diff(date(order_delivered_customer_date),date(order_estimated_delivery_
date),day ) as diff_estimated_dilevery_date
from `target_customer.orders`
order by order_id
```

Target busine... **RUN** **SAVE QUERY** **SHARE** **SCHEDULE** **MORE**

```
29 Limit 50
30 -----
31
32 5.1)
33 select date_diff(date(order_delivered_customer_date),date(order_purchase_timestamp),day) as dilever_time,
34 date_diff(date(order_delivered_customer_date),date(order_estimated_delivery_date),day) as
35 diff_estimated_dilevery_date
36 from `target_customer.orders`
37 order by order_id
```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS **EXPLORE DATA**

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	dilever_time	diff_estimated_dilevery_date				
1	7	-9				
2	16	-3				
3	8	-14				
4	6	-6				
5	25	-16				
6	7	-15				

Results per page: 50 1 - 50 of 99441

Insights: The time taken to delivery a product form the date of order will help us analyse the logistics cost incurred for every order. It will help to send the correct information about the delivery of the product through SMS and email.

Recommendations : Make necessary steps to speed the delivery by hiring multiple local delivery partners of the areas.

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

5.2) Answer

Query :

```
select geo.geolocation_state, round(avg(oi.freight_value),2) as
average_freight_value
from `target_customer.order_items` oi join `target_customer.sellers` sel
on oi.seller_id = sel.seller_id
join `target_customer.geolocation` geo
on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
group by geo.geolocation_state
order by average_freight_value desc
limit 5
```

```
7
8
9 5.2)
0 select geo.geolocation_state, round(avg(oi.freight_value),2) as average_freight_value
1 from `target_customer.order_items` oi join `target_customer.sellers` sel
2 on oi.seller_id = sel.seller_id
3 join `target_customer.geolocation` geo
4 on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
5 group by geo.geolocation_state
6 order by average_freight_value desc
7 limit 5
```

Query results

[SAVE RESULTS](#) [EX](#)

OB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTIO
v	geolocation_state	average_freight_valu			
1	CE	54.44			
2	RO	50.32			
3	PI	36.94			
4	PB	34.69			
5	AC	32.84			

5.2) answer lowest average freight value

```
select geo.geolocation_state, round(avg(oi.freight_value),2) as
lowest_average_freight_value
from `target_customer.order_items` oi join `target_customer.sellers` sel
on oi.seller_id = sel.seller_id
join `target_customer.geolocation` geo
on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
group by geo.geolocation_state
order by lowest_average_freight_value asc
```

```

limit 5
8
9 select geo.geolocation_state, round(avg(oi.freight_value),2) as lowest_average_freight_value
0 from `target_customer.order_items` oi join `target_customer.sellers` sel
1 on oi.seller_id = sel.seller_id
2 join `target_customer.geolocation` geo
3 on sel.seller_zip_code_prefix = geo.geolocation_zip_code_prefix
4 group by geo.geolocation_state
5 order by lowest_average_freight_value asc
6 limit 5
7

```

Press Alt+F1 for Acc

Query results [SAVE RESULTS](#) [EXPLORE DATA](#)

OB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
geolocation_state	lowest_average_freight_value				
1	RN		15.93		
2	SP		18.44		
3	RJ		18.93		
4	DF		18.99		
5	PR		22.11		

Job history [REFRESH](#)

Insights: The highest average freight value state is CE with 54.44 and the lowest average freight value state is RN with 15.93.

Recommendation: Carefully analyse the high freight value customers and create subscription model to avail it to reduce the transportation cost entirely on the company .Keep a minimum order value to reduce the cost of the delivery from the company side.

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

Query:

```

select
C.customer_state, avg(date_diff(date(order_delivered_customer_date), date(orde
r_purchase_timestamp), day)) as average_deliver_time
from `target_customer.orders` o inner join `target_customer.customers`
C
on o.customer_id = C.customer_id
group by C.customer_state
order by average_deliver_time desc
limit 5

```

```

5.3)
select C.customer_state, avg(date_diff(date(order_delivered_customer_date), date(order_purchase_timestamp),
day)) as average_deliver_time
from `target_customer.orders` o inner join `target_customer.customers` C
on o.customer_id = C.customer_id
group by C.customer_state
order by average_deliver_time desc
limit 5

```

Press Alt+F1 for Accessibility Q

ery results

SAVE RESULTS

EXPLORE DATA

INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
customer_state	average_deliver_time				
RR	29.34146341463...				
AP	27.17910447761...				
AM	26.35862068965...				
AL	24.50125944584...				
PA	23.72515856236...				

```

select
C.customer_state, round(avg(date_diff(date(order_delivered_customer_date), date(order
_purchase_timestamp), day)), 2) as average_deliver_time
from `target_customer.orders` o inner join `target_customer.customers` C
on o.customer_id = C.customer_id
group by C.customer_state
order by average_deliver_time asc
limit 5

```

Target busine...

RUN

SAVE QUERY

SHARE

SCHEDULE

```

8
9 5.3) lowest_average_time
0 select C.customer_state, round(avg(date_diff(date(order_delivered_customer_date), date
(order_purchase_timestamp), day)), 2) as average_deliver_time
1 from `target_customer.orders` o inner join `target_customer.customers` C
2 on o.customer_id = C.customer_id
3 group by C.customer_state
4 order by average_deliver_time asc
5 limit 5
6

```

Pres:

query results

SAVE RESULTS



OB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECU
customer_state	average_deliver_time				
1 SP	8.7				
2 PR	11.94				
3 MG	11.95				
4 DF	12.9				
5 SC	14.91				

Insights : The highest deliver time of the state gives us the no of days taken to deliver the product. We could focus on the measures to increase reduce the delivery time. Since delivery includes various factors so we need to discuss with the delivery vendor to improve the delivery. RR state has highest average delivery time taken and SP state has lowest average delivery time taken.

Recommendations: To engage the customer, regular updates of the order can be sent to increase the customer delivery experience.

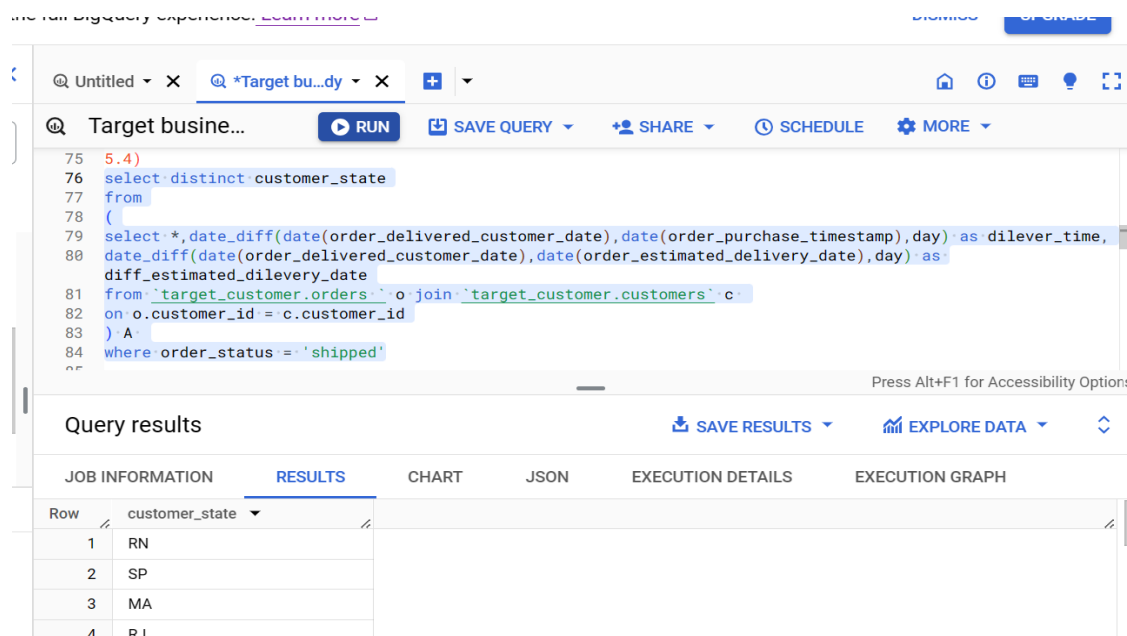
5.4) Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

5.4) Answer

Query :

```
select distinct customer_state
from
(
select
*,date_diff(date(order_delivered_customer_date),date(order_purchase_timestamp),day) as dilever_time,
date_diff(date(order_delivered_customer_date),date(order_estimated_delivery_date),day) as diff_estimated_dilevery_date
from `target_customer.orders` o join `target_customer.customers` c
on o.customer_id = c.customer_id
) A
where order_status = 'shipped'
```



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

```
5.4)
select distinct customer_state
from
(
select
*,date_diff(date(order_delivered_customer_date),date(order_purchase_timestamp),day) as dilever_time,
date_diff(date(order_delivered_customer_date),date(order_estimated_delivery_date),day) as diff_estimated_dilevery_date
from `target_customer.orders` o join `target_customer.customers` c
on o.customer_id = c.customer_id
) A
where order_status = 'shipped'
```

The query results are displayed in a table with the following columns: Row, customer_state.

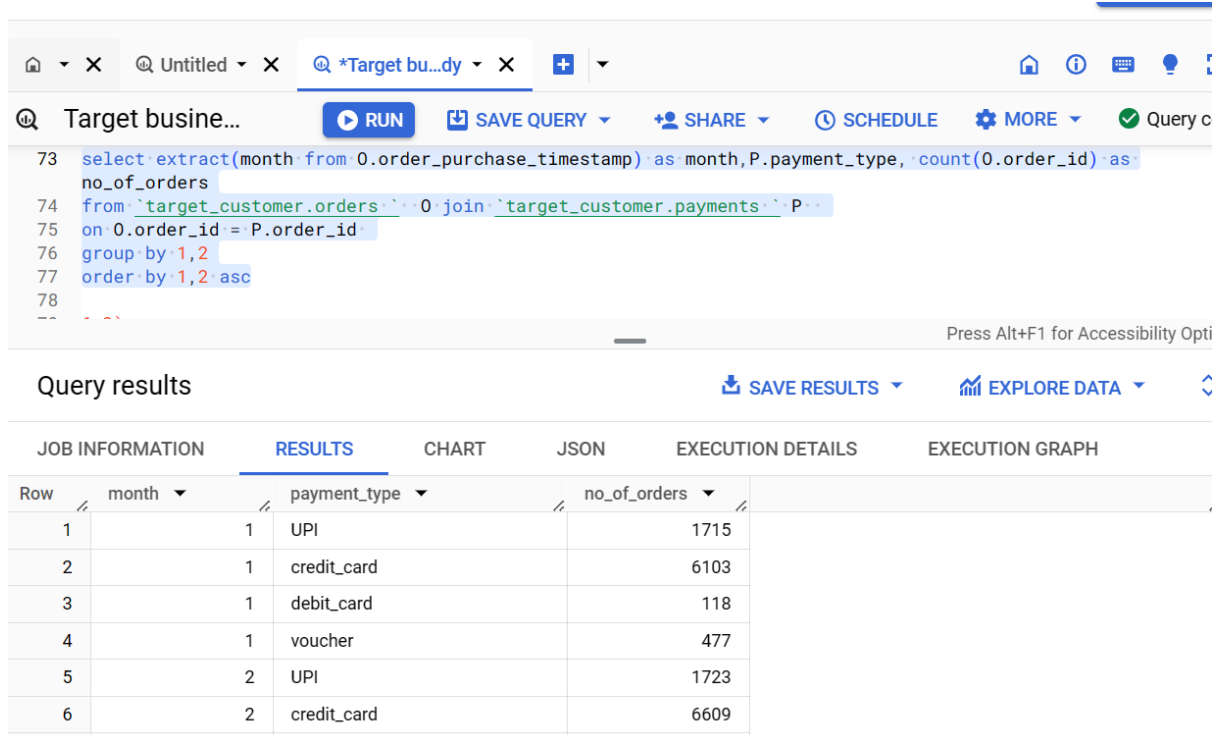
Row	customer_state
1	RN
2	SP
3	MA
4	RJ
5	BA

Insights : The top five state with fastest delivery than estimated delivery are RN,SP,MA,RJ,BA. The fast delivery than estimated delivery shows that the availability of the products in the near by ware house are high.

6. Analysis based on the payments:

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

```
select extract(month from O.order_purchase_timestamp) as month, P.payment_type,
count(O.order_id) as no_of_orders
from `target_customer.orders` O join `target_customer.payments` P
on O.order_id = P.order_id
group by 1,2
order by 1,2 asc
```



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

```
73 select extract(month from O.order_purchase_timestamp) as month, P.payment_type, count(O.order_id) as
74 no_of_orders
75 from `target_customer.orders` O join `target_customer.payments` P
76 on O.order_id = P.order_id
77 group by 1,2
78 order by 1,2 asc
```

The query results are displayed in a table with the following columns: Row, month, payment_type, and no_of_orders.

Row	month	payment_type	no_of_orders
1	1	UPI	1715
2	1	credit_card	6103
3	1	debit_card	118
4	1	voucher	477
5	2	UPI	1723
6	2	credit_card	6609

Insights : From the result, we can infer that more no of orders are placed using credit cards and there is a considerable transaction in UPI . we can understand the different payment type used by the customers and most of them are prepaid.

Recommendations:

- 1) Tie up with the banks to issue of credit card and offer some discount for the customer to encourage easy purchase.
- 2) To maintain the proper transaction history to help return the refund to the customers in case of return.
- 3) Since it is financial data, high data security must be required and follow regulations set by financial institution.

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

```
select P.payment_installments, count(O.order_id) as no_of_orders
from `target_customer.orders` O join `target_customer.payments` P
on O.order_id = P.order_id
group by P.payment_installments
order by 1
```

Target busine...

RUN

133

134 6.2) to find the number of orders placed based upon payment installments

135

136

137 select P.payment_installments, count(O.order_id) as no_of_orders

138 from `target_customer.orders` O join `target_customer.payments` P

139 on O.order_id = P.order_id

140 group by P.payment_installments

141 order by 1

142

Press Alt

Query results

SAVE RESULTS

EXI

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTIO

Row	payment_installment	no_of_orders
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
-	-	-

Results per page: 50 1 - 24 of 24

Insights: No of orders placed purchased by the customers using EMI options. Most of the product value are paid within three months of purchase. Since monthly instalments are carried out, we must follow SOP adherence to banks.

Recommendations: Since customers use various banks, we need to have multiple banks onboard to provide necessary financial services to the customers. Access management and Authentication should be strict to avoid any discrepancy.