Business Case: Target SQL

Context:

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analysing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analysing the given dataset to extract valuable insights and provide actionable recommendations.

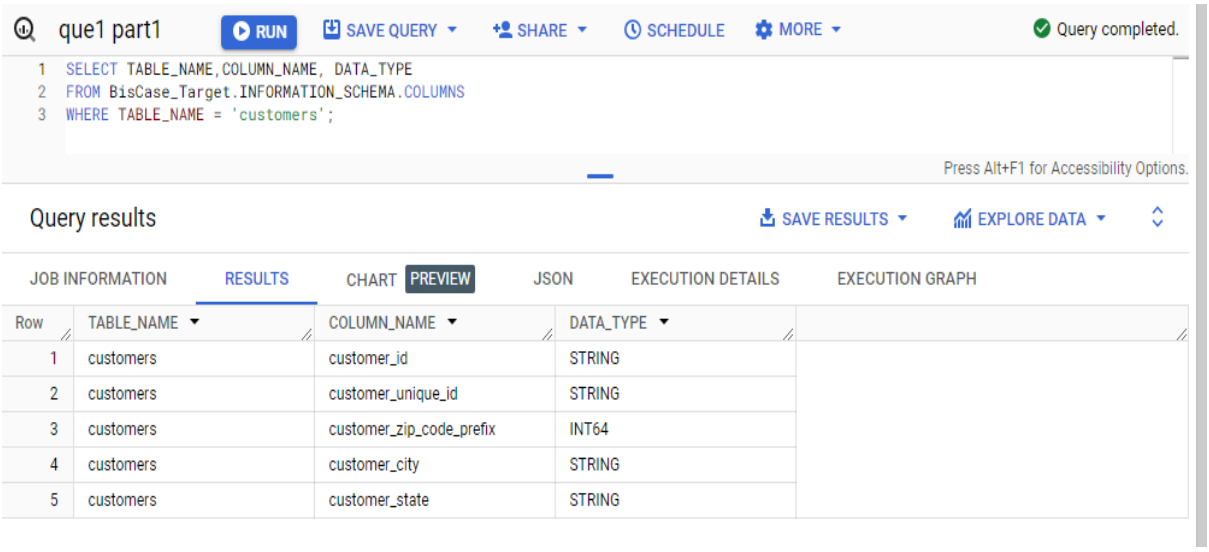
1. the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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

Query:

```
SELECT TABLE_NAME,COLUMN_NAME, DATA_TYPE
FROM BisCase_Target.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers';
```

Output:



Row	TABLE_NAME	COLUMN_NAME	DATA_TYPE
1	customers	customer_id	STRING
2	customers	customer_unique_id	STRING
3	customers	customer_zip_code_prefix	INT64
4	customers	customer_city	STRING
5	customers	customer_state	STRING

Insights:

- Most of the column in ‘customers’ table is assigned with ‘string’ data type to store text data .

Recommendation:

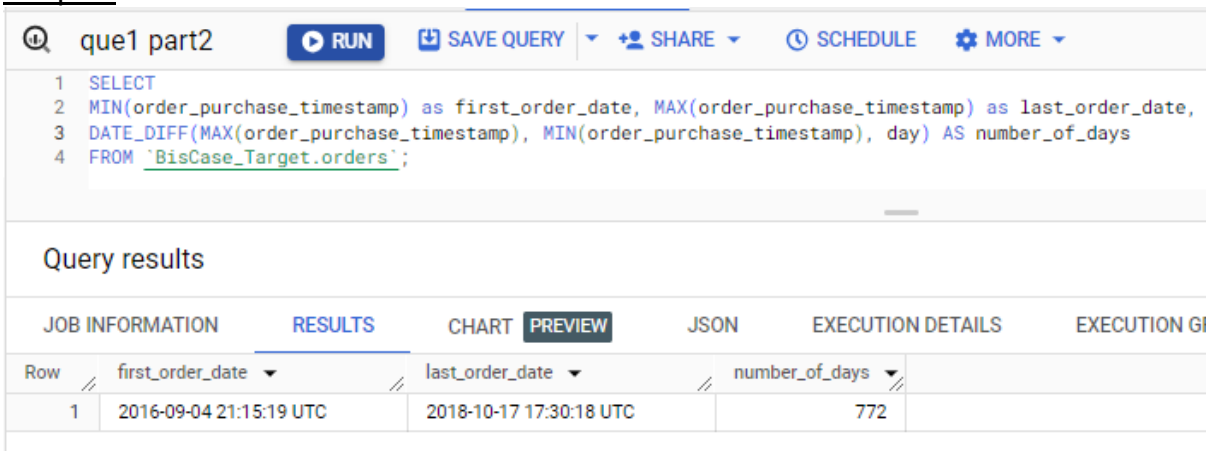
- ‘Varchar’ data type can be used for column ’customer_id’ and ‘customer_unique_id’ , that can help on entering the data limited to alpha-numeric values in these column , also we limit the number of character in this columns as ids mostly contain exact number of character.

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

Query:

```
SELECT
MIN(order_purchase_timestamp) as first_order_date, MAX(order_purchase_timestamp) as last_order_date,
DATE_DIFF(MAX(order_purchase_timestamp), MIN(order_purchase_timestamp), day) AS number_of_days
FROM `BisCase_Target.orders`;
```

Output:



Insights:

- Approx. 2 years or 772 days data is data present in ‘orders’ table. From 2016 ,4th quarter to 2018 ,4th quarter.
- Which can be useful to get the insights on seasonality change in number of orders.

Recommendations:

- It can be use to calculate the profit made in this time of duration . and make the further prediction for making profit in upcoming years

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

Query:

```
SELECT COUNT(DISTINCT geolocation_city) AS number_of_cities,
```

```
COUNT(DISTINCT geolocation_state) AS number_of_states
FROM `BisCase_Target.geolocation`;
```

Output:

que1 part3 RUN SAVE QUERY SHARE SCHEDULE MORE

```
SELECT COUNT(DISTINCT geolocation_city) AS number_of_cities,
COUNT(DISTINCT geolocation_state) AS number_of_states
FROM `BisCase_Target.geolocation`;
```

Query results

JOB INFORMATION RESULTS CHART PREVIEW JSON EXECUTION DETAILS EXECUTION GI

Row	number_of_cities	number_of_states
1	8011	27

Insights:

- The distribution of the customer cross Brazil is in 8011 cities and 27 states .
- These insight can help to provide how far company has successfully expanded. Which can promote the company reputation and help to expand other places as well.

Recommendations:

- On checking the customers orders behaviour for in any specific city or state , we can use the information to increase the customers in that area or keep the the sufficient stock for the items sold frequently in that area, which can help to reduce delivery time and reduce cost of delivery .

2. In-depth Exploration:

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,
COUNT(*) AS number_of_orders
FROM `BisCase_Target.orders`
GROUP BY year
ORDER BY year;
```

Output:

que2 part1 RUN SAVE QUERY SHARE SCHEDULE MORE This query will process 776.88 KB when ru

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
COUNT(*) AS number_of_orders
FROM BisCase_Target.orders
GROUP BY year
ORDER BY year;
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS CHART PREVIEW JSON EXECUTION DETAILS EXECUTION GRAPH

Row	year	number_of_orders
1	2016	329
2	2017	45101
3	2018	54011

Insights:

- In the span of 2 years the number of customers has increased. Where as orders increased by approx. 40k in 2016-2017 , only approx. 10k orders increased in 2017-2018 .

Recommendations:

- Company can keep record of frequently brought item in there site and keep that in stock , so that they don't loose customers.
- Company can focus on being competitive in terms of price with others company from which it is loosing its customers.

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

Query:

```
SELECT
EXTRACT ( YEAR FROM order_purchase_timestamp) AS order_year,
format_datetime("%b",order_purchase_timestamp) AS month_name,
COUNT(*) AS number_of_orders
FROM `BisCase_Target.orders`
GROUP BY order_year,month_name
ORDER BY month_name, number_of_orders desc;
```

Output:

que2 part2 RUN SAVE QUERY SHARE SCHEDULE

```
SELECT
EXTRACT ( YEAR FROM order_purchase_timestamp) AS order_year,
format_datetime("%b",order_purchase_timestamp) AS month_name,
COUNT(*) AS number_of_orders
FROM `BisCase_Target.orders`
GROUP BY order_year,month_name
ORDER BY month_name, number_of_orders desc;
```

Query results

JOB INFORMATION RESULTS CHART PREVIEW JSON EXECUTION

Row	order_year	month_name	number_of_orders
1	2018	Apr	6939
2	2017	Apr	2404
3	2018	Aug	6512
4	2017	Aug	4331
5	2017	Dec	5673
6	2016	Dec	1
7	2018	Feb	6728
8	2017	Feb	1780
9	2018	Jan	7269
10	2017	Jan	800

Insights:

- There is spike in number of order before in November in 2016 and 2017 and January month in 2017 and 2018 .which show that orders increase in cyber week and new year.

Recommendations:

- Company can take this opportunity to increase the customer , by providing more offers and sales , when eligible customers are looking for items.

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
    1. 0-6 hrs : Dawn
    2. 7-12 hrs : Mornings
    3. 13-18 hrs : Afternoon
    4. 19-23 hrs : Night

Query:

```
with cte as
 (select order_id,extract(hour FROM order_purchase_timestamp ) as hour
from `BisCase_Target.orders`)
```

```
select
 case when hour between 0 and 6 then "Dawn"
       when hour between 7 and 12 then "Morning"
       when hour between 13 and 18 then "Afternoon"
       else "Night"
 end TIME,
count(order_id) as number_of_orders
from cte
group by TIME
order by number_of_orders desc;
```

Output:

que2 part3 RUN SAVE QUERY SHARE SCHEDULE MORE

```
with cte as
 (select order_id,extract(hour FROM order_purchase_timestamp ) as hour
from `BisCase_Target.orders`)
select
 case when hour between 0 and 6 then "Dawn"
      when hour between 7 and 12 then "Morning"
      when hour between 13 and 18 then "Afternoon"
      else "Night"
 end TIME,
 count(order_id) as number_of_orders
from cte
group by TIME
order by number_of_orders desc;
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | E

Row	TIME	number_of_orders
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

- This shows that mostly customers place order in the afternoon time of the day. Which means there are high chances of customer checking website at this durations of day.

Recommendations:

- Company can take this opportunity to launch the new product at this time to increase the visibility of item , or launch the offer on item which is needs to clear from stock.

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

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

Query:

```
SELECT c.customer_state as state,
format_datetime('%b',order_purchase_timestamp) AS month,
count(order_id) AS number_of_order,
dense_rank() over (partition by c.customer_state order by count(order_id) desc) as rank1
FROM BisCase_Target.orders AS o
JOIN BisCase_Target.customers AS c
ON o.customer_id = c.customer_id
group by state,month
order by rank1;
```

Output:

que3 part1 RUN SAVE QUERY SHARE SCHEDULE MORE

```
1 SELECT c.customer_state as state,
2 format_datetime('%b',order_purchase_timestamp) AS month,
3 count(order_id) AS number_of_order,
4 dense_rank() over (partition by c.customer_state order by count(order_id) desc) as rank1
5 FROM BisCase_Target.orders AS o
6 JOIN BisCase_Target.customers AS c
7 ON o.customer_id = c.customer_id
8 group by state,month
9 order by rank1;
```

Query results SAV

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

Row	state	month	number_of_order	rank1
1	AL	Apr	51	1
2	RJ	May	1321	1
3	AP	Jan	11	1
4	AP	May	11	1
5	PI	May	56	1
6	PR	Aug	556	1
7	RN	Jul	56	1
8	SE	Aug	43	1
9	SE	Mar	43	1
10	MS	Mar	79	1

Insights:

- Highest number of orders made in August month from 4 states(PR,SE,SP,RS)
- Second highest number of orders made in May month from 9 states(MT,RJ,TO,GO,ES,AP,PI,SC,AC).

Recommendations:

- Company can clear the stock in the respective state in month where the number of order is highest
- Company take this in consideration to improve the sales in others states in month of August and May.

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

Query:

```
select customer_state as state ,count(customer_id) as no_of_customers
from `BisCase_Target.customers`
group by customer_state
order by no_of_customers desc, customer_state;
```

Output:

que3 part2 RUN SAVE QUERY SHARE SCHEDULE MORE

```
1 select customer_state as state ,count(customer_id) as no_of_customers
2 from `BisCase_Target.customers`
3 group by customer_state
4 order by no_of_customers desc, customer_state;
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTI

Row	state	no_of_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

Insights:

- The Large number of customers are present in states 'São Paulo'(SP), Rio de Janeiro (RJ) and Minas Gerais(MG) . This can help to seen the state which is providing highest profit and good company reputation

Recommendations:

- Company take this opportunity to plan on opening offline store in the state which has high number of loyal customers , for smooth success of the stores.

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).

Query:

```
with cte as
(SELECT
 SUM
 (CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM
o.order_purchase_timestamp) BETWEEN 1 AND 8 THEN
  p.payment_value
     ELSE 0
   END) AS total_payment_2017,
 SUM
 (CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 AND EXTRACT(MONTH FROM
o.order_purchase_timestamp) BETWEEN 1 AND 8 THEN
  p.payment_value
     ELSE 0
   END) AS total_payment_2018
FROM BisCase_Target.payments AS p JOIN BisCase_Target.orders AS o
ON p.order_id = o.order_id
)
SELECT CONCAT(ROUND((((total_payment_2018 - total_payment_2017) /total_payment_2017) * 100), 2),"%") AS
percentage_increase_in_cost
FROM cte;
```

Output:

Insights:

- Increase in cost of orders from 2017 to 2018 apart from months sep to dec i.e 16 months is ~137% . which gives the information about increase in cost in non-festive season of years.

Recommendations:

- Company can predict the profit in upcoming years and invest as required and stock the products based on predictions.

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

Query:

```
select c.customer_state as state ,round(sum(i.price),2) as total_price, round(avg(i.price),2) as average_price
from `BisCase_Target.customers` c join `BisCase_Target.orders` o on c.customer_id=o.customer_id
join `BisCase_Target.order_items` i on o.order_id=i.order_id
group by state
order by total_price desc,average_price desc;
```

Output:

que4 part2 RUN SAVE QUERY SHARE SCHEDULE MORE

```
select c.customer_state as state ,round(sum(i.price),2) as total_price, round(avg(i.price),2) as average_price
from `BisCase_Target.customers` c join `BisCase_Target.orders` o on c.customer_id=o.customer_id
join `BisCase_Target.order_items` i on o.order_id=i.order_id
group by state
order by total_price desc,average_price desc;
```

Query results SAVE

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

Row	state	total_price	average_price
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

Insights:

- We can get the information of total amount of ordered items in any particular state from 'total_price' column and the average amount customers spends in one order in that state from 'average_price'.
- The states with highest 'total_price' are states 'São Paulo'(SP), Rio de Janeiro (RJ) and Minas Gerais(MG).
- The average order price for all the states in the range 100-200.
- It gives the insight how much a customer ready to spend in a order in any specific state .

Recommendations:

- Company can promote the products state wise who are ready to spend the average amount similar to product .
- Can recommend the products based or there state they belong.

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

Query:

```
select c.customer_state as state ,round(sum(i.freight_value),2) as total_freight, round(avg(i.freight_value),2) as average_freight
from `BisCase_Target.customers` c join `BisCase_Target.orders` o on c.customer_id=o.customer_id
join `BisCase_Target.order_items` i on o.order_id=i.order_id
group by state
order by total_freight desc, average_freight desc;
```

Output:

que4 part3 RUN SAVE QUERY SHARE SCHEDULE MORE This query will pr

```
select c.customer_state as state ,round(sum(i.freight_value),2) as total_freight, round(avg(i.freight_value),2) as average_freight
from `BisCase_Target.customers` c join `BisCase_Target.orders` o on c.customer_id=o.customer_id
join `BisCase_Target.order_items` i on o.order_id=i.order_id
group by state
order by total_freight desc,average_freight desc;
```

Press Al

Query results SAVE RESULTS EX

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

Row	state	total_freight	average_freight
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

Insights:

- We can get the information of total freight value of orders in any particular state from 'total_freight' column and the average freight amount customers paid in one order in that state from 'average_freight'.
- The states with highest 'total_freight' are states 'São Paulo'(SP), Rio de Janeiro (RJ) and Minas Gerais(MG).

Recommendations:

- By look into the 'total_freight' value company can decide on increasing year-or-year delivery charges , the state with high 'total_freight' value customers are willing to order with delivery charges.

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.

Query:

```
with cte as
(select c.customer_id, o.order_id,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,
datetime_diff(order_estimated_delivery_date,order_delivered_customer_date, day) as diff_estimated_delivery
from `BisCase_Target.customers` as c join `BisCase_Target.orders` as o on c.customer_id=o.customer_id)
select  customer_id,order_id,time_to_deliver ,
case when diff_estimated_delivery>0 then concat(diff_estimated_delivery," days late")
   when diff_estimated_delivery<0 then concat(-(diff_estimated_delivery)," days early")
   else "On time"
 End diff_estimated_delivery
 from cte
```

Output:

que5 part1 RUN SAVE QUERY SHARE SCHEDULE MORE

```
1  with cte as
2  (select c.customer_id, o.order_id,
3  datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,
4  datetime_diff(order_estimated_delivery_date,order_delivered_customer_date, day) as diff_estimated_delivery
5  from `BisCase_Target.customers` as c join `BisCase_Target.orders` as o on c.customer_id=o.customer_id)
6  select  customer_id,order_id,time_to_deliver ,
7  case when diff_estimated_delivery>0 then concat(diff_estimated_delivery," days late")
8      when diff_estimated_delivery<0 then concat(-(diff_estimated_delivery)," days early")
9      else "On time"
10  End diff_estimated_delivery
11  from cte
```

Query results

SAVE RESULTS

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

Row	customer_id	order_id	time_to_deliver	diff_estimated_delivery
1	1bccb206de9f0f25adc6871a1...	1950d777989f6a877539f5379...	30	12 days early
2	de4caa97afa80c8eeac2ff4c8d...	2c45c33d2f9cb8ff8b1c86cc28...	30	28 days late
3	70fc57eeae292675927697fe0...	65d1e226dfaeb8cdc42f66542...	35	16 days late
4	7a34a8e890765ad6f90db76d0...	635c894d068ac37e6e03dc54e...	30	1 days late
5	065d53860347d845788e041c...	3b97562c3aee8bdedcb5c2e45...	32	On time
6	0378e1381c730d4504ebc07d2...	68f47f50f04c4cb6774570cfde...	29	1 days late
7	d33e520a99eb4cfc0d3ef2b6ff...	276e9ec344d3bf029ff83a161c...	43	4 days early
8	a0bc11375dd3d8bdd0e0bfcbc...	54e1a3c2b97fb0809da548a59...	40	4 days early
9	8fe0db7abbccaf2d788689e91...	fd04fa4105ee8045f6a0139ca5...	37	1 days early
10	22c0028cdec95ad1808c1fd50...	302bb8109d097a9fc6e9cefc5...	33	5 days early

Insights:

- We can get the information how much time it has taken to deliver an order in 'time_to_deliver' column and if its delivered late or early from the estimated date in 'diff_estimated_delivery'.
- From above result me can see most of the orders are delivered before time .
- It can help to calculate or average how much time they are taking to deliver the order to there customers.

Recommendations:

- Where the orders are delivered late from the estimated date , company improve the delivery for the area where the orders are mostly delivered late , by hiring more people there , and investing in wharehouses.

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

Query:

```
SELECT
 h.customer_state AS high_state,h. average_freight_value AS high_avg_freight,
 l.customer_state AS low_state,l. average_freight_value AS low_avg_freight
FROM
(
 SELECT c.customer_state,
 ROUND(AVG(i.freight_value),2) AS average_freight_value,
 DENSE_RANK() OVER(ORDER BY(ROUND(AVG(i.freight_value),2))DESC) AS rnk1
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.order_items` AS i ON o.order_id = i.order_id
 JOIN `BisCase_Target.customers` AS c ON o.customer_id = c.customer_id
 GROUP BY c.customer_state
 ORDER BY average_freight_value DESC
 LIMIT 5
 ) AS h
JOIN
(
  SELECT c.customer_state,
```

```
 ROUND(AVG(i.freight_value),2) AS average_freight_value,
 DENSE_RANK() OVER(ORDER BY (ROUND(AVG(i.freight_value),2)))AS rnk2
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.order_items` AS i ON o.order_id = i.order_id
 JOIN `BisCase_Target.customers` AS c ON o.customer_id = c.customer_id
 GROUP BY c.customer_state
 ORDER BY average_freight_value
 LIMIT 5
) AS l
ON h.rnk1 =l.rnk2;
```

Output:

que5 part2 RUN SAVE QUERY SHARE SCHEDULE MORE

```
SELECT
 h.customer_state AS high_state,h.average_freight_value AS high_avg_freight,
 l.customer_state AS low_state,l.average_freight_value AS low_avg_freight
FROM
(
 SELECT c.customer_state,
 ROUND(AVG(i.freight_value),2) AS average_freight_value,
 DENSE_RANK() OVER(ORDER BY(ROUND(AVG(i.freight_value),2))DESC) AS rnk1
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.order_items` AS i ON o.order_id = i.order_id
 JOIN `BisCase_Target.customers` AS c ON o.customer_id = c.customer_id
 GROUP BY c.customer_state
 ORDER BY average_freight_value DESC
 LIMIT 5
 ) AS h
JOIN
(
   SELECT c.customer_state,
 ROUND(AVG(i.freight_value),2) AS average_freight_value,
 DENSE_RANK() OVER(ORDER BY (ROUND(AVG(i.freight_value),2)))AS rnk2
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.order_items` AS i ON o.order_id = i.order_id
 JOIN `BisCase_Target.customers` AS c ON o.customer_id = c.customer_id
 GROUP BY c.customer_state
 ORDER BY average_freight_value
 LIMIT 5
) AS l
ON h.rnk1 =l.rnk2;
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECU

Row	high_state	high_avg_freight	low_state	low_avg_freight
1	RR	42.98	SP	15.15
2	PB	42.72	PR	20.53
3	RO	41.07	MG	20.63
4	AC	40.07	RJ	20.96
5	PI	39.15	DF	21.04

Insights:

- Its provide the information of 5 states with highest freight price value and lowest freight price value.
- The average freight price for all the states in the range 15-43.

Recommendations:

- In the state with high ‘average_freight’ values , company can promote the free delivery membership programs, where its most likely to be purchased to save delivery charges in every order.
- Company can increase the price amount of order for free delivery for the state which have high ‘average_freight’ values, as people will likely to spend that much to make the delivery free for order.

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

Query:

```
WITH cte AS
(
 SELECT c.customer_state,
 FLOOR(AVG(DATETIME_DIFF(order_delivered_customer_date,order_purchase_timestamp, day))) AS
avg_delivery_days
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.customers` AS c
 ON o.customer_id = c.customer_id
 WHERE order_status = 'delivered' AND order_delivered_customer_date IS NOT NULL
 GROUP BY c.customer_state
 ORDER BY avg_delivery_days
)
SELECT h.customer_state AS high_avg_state,CONCAT(h.avg_delivery_days,' DAYS') AS
high_avg_delivery_days,
l.customer_state AS low_avg_state, CONCAT(l.avg_delivery_days,' DAYS') AS
low_avg_delivery_days
FROM
(
 SELECT *,
 DENSE_RANK() OVER (ORDER BY avg_delivery_days DESC) AS rnk1
 FROM cte
 ORDER BY rnk1
 LIMIT 5
) AS h
JOIN
(
 SELECT *,
 DENSE_RANK() OVER (ORDER BY avg_delivery_days) AS rnk2
 FROM cte
 ORDER BY rnk2
 LIMIT 5
) AS l
ON h.rnk1 = l.rnk2;
```

Output:

que5 part3 RUN SAVE QUERY SHARE SCHEDULE MORE

```
WITH cte AS
(
 SELECT c.customer_state,
 FLOOR(AVG(DATETIME_DIFF(order_delivered_customer_date,order_purchase_timestamp, day))) AS avg_delivery_days
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.customers` AS c
 ON o.customer_id = c.customer_id
 WHERE order_status = 'delivered' AND order_delivered_customer_date IS NOT NULL
 GROUP BY c.customer_state
 ORDER BY avg_delivery_days
)
SELECT h.customer_state AS high_avg_state,CONCAT(h.avg_delivery_days,' DAYS') AS high_avg_delivery_days,
l.customer_state AS low_avg_state, CONCAT(l.avg_delivery_days,' DAYS') AS low_avg_delivery_days
FROM
(
 SELECT *,
 DENSE_RANK() OVER (ORDER BY avg_delivery_days DESC) AS rnk1
 FROM cte
 ORDER BY rnk1
 LIMIT 5
) AS h
JOIN
(
 SELECT *,
 DENSE_RANK() OVER (ORDER BY avg_delivery_days) AS rnk2
 FROM cte
 ORDER BY rnk2
 LIMIT 5
) AS l
ON h.rnk1 = l.rnk2;
```

Query results SAVE RESUI

JOB INFORMATION RESULTS CHART PREVIEW JSON EXECUTION DETAILS EXECUTION GRAPH

Row	high_avg_state	high_avg_delivery_days	low_avg_state	low_avg_delivery_days
1	RR	28 DAYS	SP	8 DAYS
2	AP	26 DAYS	MG	11 DAYS
3	AP	26 DAYS	PR	11 DAYS
4	AM	25 DAYS	DF	12 DAYS
5	AL	24 DAYS	SC	14 DAYS

Insights:

- The above query gives the information on the 5 states with highest average delivery days in 'high_avg_delivery_days' and with lowest average delivery days in 'low_avg_delivery_days'.
- The average delivery days for all states is in range of 28-8 days

Recommendations:

- Company can hire more delivery partners and build warehouse to reduce the average delivery days in states with high_avg_delivery_days'. Which will also help to improve compary reputation.

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

Query:

```
WITH cte AS
```

```
(
 SELECT c.customer_state,
 AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY)) AS
avg_delivery,
 DENSE_RANK() OVER (ORDER BY
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY))) AS
rnk
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.customers` AS c
 ON o.customer_id = c.customer_id
 WHERE o.order_delivered_customer_date IS NOT NULL AND o.order_estimated_delivery_date IS
NOT NULL
 GROUP BY c.customer_state
)
SELECT customer_state AS STATE, CONCAT(FLOOR(-(avg_delivery)),"  days earlier") AS
AVG_DIFF_IN_DELIVERY_DAYS
FROM cte
WHERE rnk <= 5
ORDER BY avg_delivery;
```

Output:

que5 part4 RUN SAVE QUERY SHARE SCHEDULE MORE This c

```
WITH cte AS
(
 SELECT c.customer_state,
 AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY)) AS avg_delivery,
 DENSE_RANK() OVER (ORDER BY AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY))) AS rnk
 FROM `BisCase_Target.orders` AS o JOIN `BisCase_Target.customers` AS c
 ON o.customer_id = c.customer_id
 WHERE o.order_delivered_customer_date IS NOT NULL AND o.order_estimated_delivery_date IS NOT NULL
 GROUP BY c.customer_state
)
SELECT customer_state AS STATE, CONCAT(FLOOR(-(avg_delivery)),"  days earlier") AS AVG_DIFF_IN_DELIVERY_DAYS
FROM cte
WHERE rnk <= 5
ORDER BY avg_delivery;
```

Query results SAVE RESULTS

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

Row	STATE	AVG_DIFF_IN_DELIVERY_DAYS
1	AC	19 days earlier
2	RO	19 days earlier
3	AP	18 days earlier
4	AM	18 days earlier
5	RR	16 days earlier

Insights:

- The above query gives the information on the 5 states highest average number of days where ordered delivered before estimated delivery date in 'AVG_DIFF_IN_DELIVERY_DAYS' column.
- The least days consumed to delivery the orders is in state Acre(AC) and Rondônia (RO).

Recommendations:

- The factors in these states which leads to the early delivery can help to make changes and improve the delivery in order states as well.

6. Analysis based on the payments:

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

Query:

```
SELECT
 FORMAT_TIMESTAMP('%Y-%m', o.order_purchase_timestamp) AS month,
 p.payment_type,
 COUNT(DISTINCT o.order_id) AS number_of_orders
FROM `BisCase_Target.orders` AS o
JOIN `BisCase_Target.payments` AS p
ON o.order_id = p.order_id
GROUP BY month, p.payment_type
ORDER BY month, number_of_orders desc;
```

Output:

que6 part1 RUN SAVE QUERY SHARE SCHEDULE

```
SELECT
 FORMAT_TIMESTAMP('%Y-%m', o.order_purchase_timestamp) AS month,
 p.payment_type,
 COUNT(DISTINCT o.order_id) AS number_of_orders
FROM `BisCase_Target.orders` AS o
JOIN `BisCase_Target.payments` AS p
ON o.order_id = p.order_id
GROUP BY month, p.payment_type
ORDER BY month, number_of_orders desc;
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION

Row	month	payment_type	number_of_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	253
3	2016-10	UPI	63
4	2016-10	voucher	11
5	2016-10	debit_card	2
6	2016-12	credit_card	1
7	2017-01	credit_card	582
8	2017-01	UPI	197
9	2017-01	voucher	33
10	2017-01	debit_card	9

Insights:

- the above query gives the information number of order places with different payment type in each month.
- Most of the orders paid with ‘credit_card’ almost in every month , which maybe be because the offers provide on using ‘credit_card’ as payment method.

Recommendations:

- Company can present this data in improve the contracts with banks to promote there ‘credit_card’ or offer to promote there debit cards as well or any other mode of payment .

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

Query:

```
SELECT payment_installments,
 COUNT(DISTINCT order_id) AS number_of_orders
FROM `BisCase_Target.payments`
GROUP BY payment_installments
ORDER BY number_of_orders desc;
```

Output:

que6 part2 RUN SAVE QUERY

```
SELECT payment_installments,
 COUNT(DISTINCT order_id) AS number_of_orders
FROM `BisCase_Target.payments`
GROUP BY payment_installments
ORDER BY number_of_orders desc;
```

Query results

JOB INFORMATION RESULTS CHART PREVIEW

Row	payment_installment	number_of_orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	10	5315
6	5	5234
7	8	4253
8	6	3916
9	7	1623
10	9	644

Insights:

- The above query shows the number of orders with places against the number of payment instalment
- From the above results we can see most of the highest number of payment made in 1 instalment , which gives the insights that most of the order places in website are not high price products , where customers are able to pay in one time .
- There is also the sufficient numbers of orders in 2 ,3,4,10 , 5 number of instalment , with means the people maybe buying the high priced items from site as well.

Recommendations:

- Company can provide offers on paying in instalments and promote this , so that the customer buy more of high priced item from the site.