TARGET-SQL

I. 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.

Ans :

```
SELECT column_name,data_type
FROM `Target_SQL.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers';
```

Query results

SAVE RESULTS | EXPLORE DATA

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

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 :

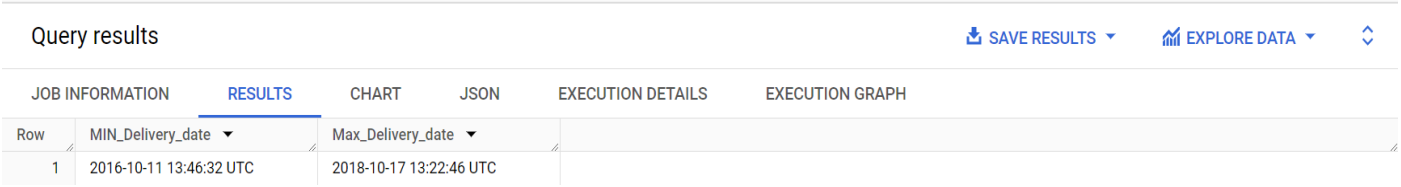
- The question focuses on the "customers" table, indicating an interest in understanding the data types of its columns.

Recommendations : NA

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

Ans:

```
SELECT
       MIN(order_delivered_customer_date) AS MIN_Delivery_date,
       MAX(order_delivered_customer_date) AS Max_Delivery_date
FROM
       `Target_SQL.orders`
WHERE
       order_delivered_customer_date is not null;
```



Insights :

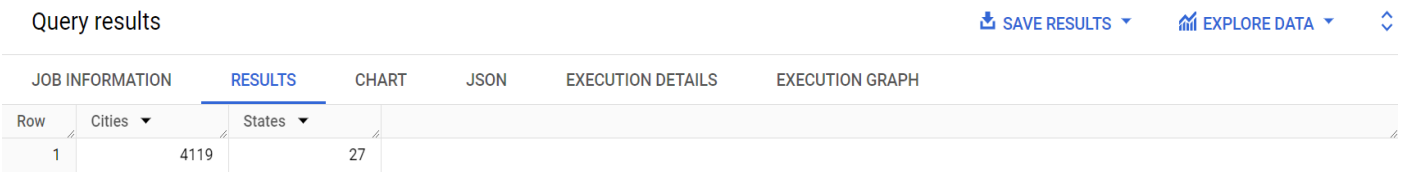
- 'MIN_Delivery_date' It indicates when the first order in the dataset was made.
- 'Max_Delivery_date' It indicates when the most recent order in the dataset was made.

Recommendations : NA

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

Ans:

```
SELECT
       COUNT(DISTINCT C.customer_city) AS Cities,
       COUNT(DISTINCT C.customer_state) AS States
FROM `Target_SQL.orders` O INNER JOIN `Target_SQL.customers` C
       ON O.customer_id  =  C.customer_id;
```



Insights :

- The query aims to count the cities and states of customers who placed orders during a given period.

Recommendations : NA

2. In-depth Exploration.

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

Ans :

```
SELECT
       Year,
       No_of_order_placed_year
FROM
```

```
        (SELECT
              FORMAT_DATE('%Y', order_purchase_timestamp)AS Year,
              COUNT(FORMAT_DATE ('%Y', order_purchase_timestamp)) AS
              No_of_order_placed_year
        FROM `Target_SQL.orders`
        GROUP BY
              FORMAT_DATE ('%Y', order_purchase_timestamp)
        ) Table_1

ORDER BY 1;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Year	No_of_order_placed_year
1	2016	329
2	2017	45101
3	2018	54011

Insights :

- The analysis of the 'No_of_order_placed_year' column shows a consistent increase in the number of orders placed over the year, from 329 in 2016 to 54011 in 2018.

Recommendations :

- To evaluate and ensure that the business has the necessary infrastructure and capacity to handle the increasing number of orders.
- This includes inventory management, production capabilities, and logistics.

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

Ans :

```
SELECT
      Table_1.Year_month,
      Table_1.monthly_order
FROM
      (SELECT
            FORMAT_DATE ('%Y-%m',order_purchase_timestamp) as Year_month,
            COUNT(FORMAT_DATE ('%Y-%m',order_purchase_timestamp)) as monthly_order
      FROM `Target_SQL.orders`
      GROUP BY
            FORMAT_DATE ('%Y-%m',order_purchase_timestamp)) Table_1

ORDER BY 1;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Year_month	monthly_order
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026

Insights :

- The number of orders spikes to 4026. This a seasonality pattern related to increased demand during festive periods.
- The numbers progressively rise from 4 in September(2016-09) to 4026 in July (2017-07). This overall growth is positive for the business.

Recommendations :

- Leverage slower months such as August and September and October for targeted promotional campaigns. Offer discounts, promotions, or exclusive deals to stimulate demand and attract customer during these periods.

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

a. 0-6 hrs : Dawn
b. 7-12 hrs : Mornings
c. 13-18 hrs : Afternoon
d. 19-23 hrs : Night

Ans :

```
WITH total AS

(SELECT
      Table_.Hours_in_24_format,
      Table_.orderss

FROM
      (SELECT
            FORMAT_DATE('%H',order_purchase_timestamp) AS Hours_in_24_format,
            COUNT(order_purchase_timestamp) AS orderss
      FROM `Target_SQL.orders`
      GROUP BY FORMAT_DATE('%H',order_purchase_timestamp)) Table_
ORDER BY 1)
```

```sql
SELECT
       Hours,
       Total_orders
FROM
       (SELECT
              DISTINCT Hours,Total_orders
       FROM
              (SELECT *,
                     SUM(total.orderss)OVER() AS Total_orders,'1. 0-6 hrs : Dawn'    AS
                     Hours
              FROM
                     total
              WHERE Hours_in_24_format BETWEEN '00' AND '06')


UNION ALL

SELECT
       DISTINCT Hours,Total_orders
FROM
       (SELECT *,
              SUM(total.orderss)OVER() AS Total_orders,'2. 7-12 hrs : Mornings' AS
              Hours,
       FROM
              total
       WHERE
              Hours_in_24_format BETWEEN '07' AND '12')

UNION ALL

SELECT
       DISTINCT Hours,Total_orders
FROM
       (SELECT *,
              SUM(total.orderss)OVER() AS Total_orders,'3. 13-18 hrs : Afternoon' AS
              Hours
       FROM
              total
       WHERE
              Hours_in_24_format BETWEEN '13' AND '18')

UNION ALL

SELECT
       DISTINCT Hours,Total_orders
FROM
       (SELECT *,
              SUM(total.orderss)OVER() AS Total_orders,'4. 19-23 hrs : Night' AS Hours
       FROM
              total
       WHERE
              Hours_in_24_format BETWEEN '19' AND '23'))

ORDER BY 1;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Hours	Total_orders
1	1. 0-6 hrs : Dawn	5242
2	2. 7-12 hrs : Mornings	27733
3	3. 13-18 hrs : Afternoon	38135
4	4. 19-23 hrs : Night	28331

Insights :

- The ‘afternoon’ time slot (13:00 – 18:00) has the highest total orders (38135).
- This indicates that Brazilian customers predominantly place during the afternoon hours.

Recommendations :

- Given the peak in orders during the Afternoon, businesses should focus on optimizing operations, including staffing and inventory, to meet the increased demand during these hours.
- Consider offers or incentives specifically designed for the Afternoon hours to increase on customer activity.

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

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

Ans :

```sql
SELECT
    Year,
    Month,
    State,
    No_of_Orders

FROM
    (SELECT
              FORMAT_DATE('%Y',O.order_purchase_timestamp)AS  Year  ,
              FORMAT_DATE ('%m',O.order_purchase_timestamp) AS Month,
              C.customer_state as State,
              COUNT(O.order_purchase_timestamp) AS No_of_Orders
    FROM
       `Target_SQL.orders` O INNER JOIN `Target_SQL.customers` C ON
       O.customer_id  =  C.customer_id
    GROUP BY
              C.customer_state,
              FORMAT_DATE ('%Y',order_purchase_timestamp),
              FORMAT_DATE ('%m',order_purchase_timestamp))table_1

ORDER BY
    1,2,3,4;
```

Query results

SAVE RESULTS EXPLORE DATA

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

Row	Year	Month	State	No_of_Orders
1	2016	09	RR	1
2	2016	09	RS	1
3	2016	09	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

Insights :

- The outputs are allows to observe the trends in the number of orders placed month-on-month. Look for patterns such as growth, decline, or seasonality.

Recommendations :

- Develop targeted marketing campaigns for states with those with lower order volumes.
- Utilize geotargeted advertising to reach customers in specific states.

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

Ans :

```
SELECT
    customer_state AS Customer_State,
    COUNT(customer_id) AS No_of_Customers
FROM
    `Target_SQL.customers`
GROUP BY
    customer_state
ORDER BY
    2 DESC;
```

Query results

SAVE RESULTS EXPLORE DATA

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

Row	Customer_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 distribution of customers across different states. It shows the count of customers in each state, helping identify states with higher or lower customer bases.

Recommendations :

- Collect customer feedback to each region. Understanding the needs and expectations of customers in different states can guide product development, service improvements.

- Run targeted online ads to increase brand awareness and attract new customers in those lower customer counts regions.

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

A. 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.

Ans :

```
WITH Total AS

(SELECT
    FORMAT_DATE('%Y', O.order_purchase_timestamp) AS year_,
    SUM(P.payment_value) AS total_payment,

FROM
    `Target_SQL.payments` P INNER JOIN `Target_SQL.orders` O ON P.order_id = O.order_id

WHERE

    EXTRACT(YEAR FROM O.order_purchase_timestamp) IN (2017, 2018)
    AND EXTRACT(MONTH FROM O.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY
    FORMAT_DATE('%Y', O.order_purchase_timestamp))


SELECT
     ROUND(((T1.total_payment - T2.Total_payment) /T2.Total_payment)*100,4) AS
     Percentage_of_increase
FROM
    total T1 INNER JOIN total T2 ON T1.year_ <> T2.year_
WHERE
    T1.year_='2018' AND T2.year_ = '2017';
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Percentage_of_increase
1	136.9769

Insights:

- The insights of the question is how the total cost of orders has changed between the specified years and months.

Recommendations:

- The primary factors contributing to the increase in order costs. Understanding the drivers can guide targeted business decisions
- Examine how the cost increase impacts overall profitability. Ensure that the rise in costs is aligned with revenue growth and does not adversely affect margins

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

Ans :

```
SELECT
    C.customer_state,
    ROUND(AVG(P.payment_value),2) AS Avg_Value,
    ROUND(SUM(P.payment_value),2) AS Total_Value
FROM `Target_SQL.customers` C
      INNER JOIN `Target_SQL.orders` O ON C.customer_id = O.customer_id
      INNER JOIN `Target_SQL.payments` P ON O.order_id = P.order_id
GROUP BY
    C.customer_state

ORDER BY
    1;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Avg_Value	Total_Value
1	AC	234.29	19680.62
2	AL	227.08	96962.06
3	AM	181.6	27966.93
4	AP	232.33	16262.8
5	BA	170.82	616645.82
6	CE	199.9	279464.03
7	DF	161.13	355141.08
8	ES	154.71	325967.55
9	GO	165.76	350092.31
10	MA	198.86	152523.02

Insights :

- The ‘Avg_Value’ provides the average payment value for each state. This can give you an idea of the typical order amount for customers in different states.
- The ‘Total_Value’ shows the total payment value for each state. This helps identify the states contributing the most to the overall revenue.

Recommendations :

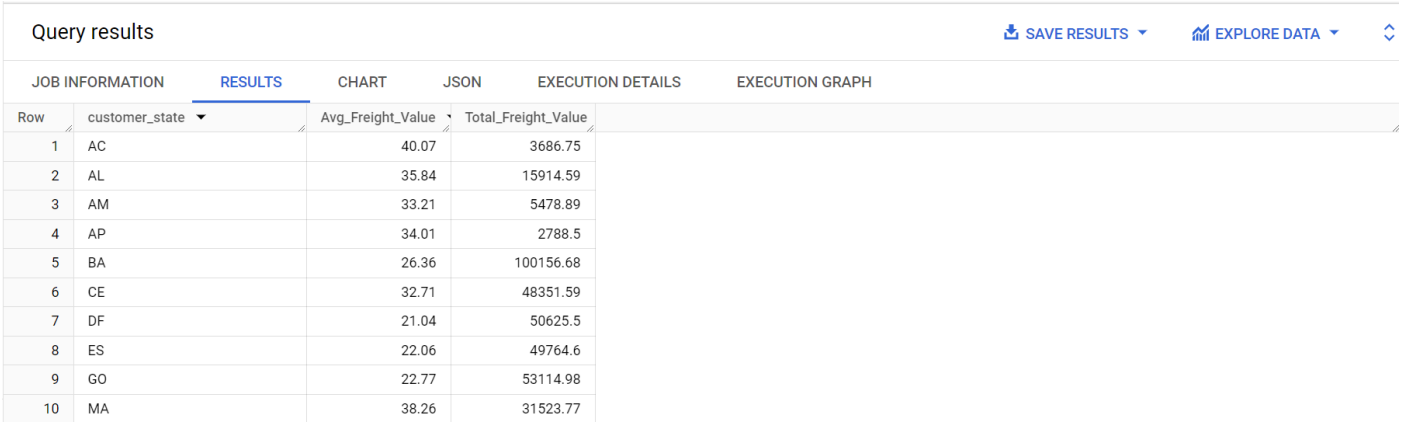
- Concentrate marketing efforts in states where customers spend more on average.

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

Ans :

```sql
SELECT
    C.customer_state,
    ROUND(AVG(OI.freight_value),2) AS Avg_Freight_Value,
    ROUND(SUM(OI.freight_value),2) AS Total_Freight_Value
FROM `Target_SQL.customers` C
      INNER JOIN `Target_SQL.orders` O ON C.customer_id = O.customer_id
      INNER JOIN `Target_SQL.order_items` OI ON O.order_id = OI.order_id
GROUP BY
    C.customer_state

ORDER BY
    1;
```



Query results

SAVE RESULTS EXPLORE DATA

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

Row	customer_state	Avg_Freight_Value	Total_Freight_Value
1	AC	40.07	3686.75
2	AL	35.84	15914.59
3	AM	33.21	5478.89
4	AP	34.01	2788.5
5	BA	26.36	100156.68
6	CE	32.71	48351.59
7	DF	21.04	50625.5
8	ES	22.06	49764.6
9	GO	22.77	53114.98
10	MA	38.26	31523.77

Insights:

- The Total_Freight_Value represents the sum of freight values for each state. This gives you the overall cost of shipping orders to each state.
- The Avgerage_Freight_Value provides the average freight value for each state. This metric can give you an idea of the typical cost of shipping an order to different states.

Recommendations :

- states with higher average freight values, explore optimizing shipping strategies or partnering with specific carriers to reduce costs.
- Analyze products frequently ordered together in states with high total freight values.

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

A. 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.

Ans:

```
SELECT
      order_id,
       DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) AS
       time_to_deliver,
      DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day) AS
      diff_estimated_delivery,
FROM
    `Target_SQL.orders`
WHERE
       order_status = 'delivered'
ORDER BY 1;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	-8
2	00018f77f2f0320c557190d7a1...	16	-2
3	000229ec398224ef6ca0657da...	7	-13
4	00024acbcdf0a6daa1e931b03...	6	-5
5	00042b26cf59d7ce69dfabb4e...	25	-15
6	00048cc3ae777c65dbb7d2a06...	6	-14
7	00054e8431b9d7675808bcb8...	8	-16
8	000576fe39319847cbb9d288c...	5	-15
9	0005a1a1728c9d785b8e2b08b...	9	0
10	0005f50442cb953dcd1d21e1f...	2	-18

Insights:

- The time_to_deliver column represents the actual time taken to deliver each order from the purchase date. It provides insights into the efficiency of the delivery process and the average time it takes for orders to reach customers.

- The diff_estimated_delivery column calculates the variance between the estimated and actual delivery dates. This metric highlights how accurate the initial delivery estimates are and the consistency in meeting those estimates.

Recommendations:

- Analyze orders with longer time_to_deliver values to identify operational inefficiencies. Optimize logistics and fulfillment processes to reduce delivery times and enhance overall efficiency.
- Focus on reducing delivery times and meeting estimated delivery dates to enhance overall customer satisfaction.

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

Ans :

```
WITH total AS

(SELECT
      C.customer_state,
      AVG(OI.freight_value) AS avg_freight
FROM
      `Target_SQL.customers` C INNER JOIN `Target_SQL.orders` O ON C.customer_id  =
      O.customer_id
      INNER JOIN `Target_SQL.order_items` OI ON O.order_id  =  OI.order_id
GROUP BY
      C.customer_state
)

(SELECT
      customer_state AS state,
      'Highest' AS Type_of_freight_value,
      avg_freight AS Avg_freight_value
FROM
       total
ORDER BY 3 DESC
LIMIT 5)

UNION ALL

(SELECT
      customer_state AS state,
      'Lowest' AS Type_of_freight_value,
      avg_freight AS Avg_freight_value
FROM
       total
ORDER BY 3 ASC
LIMIT 5);
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	Type_of_freight_value	Avg_freight_value
1	RR	Highest	42.98442307692...
2	PB	Highest	42.72380398671...
3	RO	Highest	41.06971223021...
4	AC	Highest	40.07336956521...
5	PI	Highest	39.14797047970...
6	SP	Lowest	15.14727539041...
7	PR	Lowest	20.53165156794...
8	MG	Lowest	20.63016680630...
9	RJ	Lowest	20.96092393168...
10	DF	Lowest	21.04135494596...

Insights :

- Calculates the average freight value for each state and selects the top 5 states with the highest average freight values. This provides insights into states where customers, on average, incur higher shipping costs.

- The second part of the query identifies the states with the lowest average freight values. It highlights states where customers, on average, experience lower shipping costs.

Recommendations:

- Analyze the states with the highest average freight values to optimize shipping strategies. Explore options to negotiate better shipping rates, improve logistics efficiency, or consider alternative carriers.

- Optimize the distribution network to minimize shipping costs and improve delivery times.
- Use insights gained to make informed decisions and adjustments to your shipping and pricing policies.

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

Ans:

```
WITH Total AS
(
SELECT
    C.customer_state,
    ROUND(AVG(DATE_DIFF(O.order_delivered_customer_date,
              O.order_purchase_timestamp,day)),2) AS Delivery_time
FROM
    `Target_SQL.orders` O INNER JOIN `Target_SQL.customers` C
      ON O.customer_id = C.customer_id
GROUP BY
    C.customer_state
)
```

```sql
(SELECT
    Customer_State,
    'Highest' as Delivery_Time,
    Delivery_time AS Avg_Delivery_Time
FROM
    total
ORDER BY 3 DESC
LIMIT 5)

UNION ALL

(SELECT
    Customer_State,
    'Lowest' as type,
    Delivery_time AS Avg_Delivery_Time
FROM
    total
ORDER BY 3 ASC
LIMIT 5);
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Customer_State	Delivery_Time	Avg_Delivery_Time
1	RR	Highest	28.98
2	AP	Highest	26.73
3	AM	Highest	25.99
4	AL	Highest	24.04
5	PA	Highest	23.32
6	SP	Lowest	8.3
7	PR	Lowest	11.53
8	MG	Lowest	11.54
9	DF	Lowest	12.51
10	SC	Lowest	14.48

Insights:

- The first part of the query calculates the average delivery time for each state and selects the top 5 states with the highest average delivery times. This provides insights into states where, on average, customers experience longer delivery times.

- The second part of the query identifies the states with the lowest average delivery times. It highlights states where customers, on average, receive their orders more quickly.

- Differences in delivery times may be influenced by factors such as shipping infrastructure, distance from distribution centers, or carrier efficiency.

Recommendations :

- States with higher average delivery times may benefit from optimizing the logistics and distribution network.
- Consider adjusting the location of distribution centers or optimizing shipping routes to reduce delivery times.
- Evaluate the performance of carriers operating in states with varying average delivery times.

D. 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.

Ans :

```
WITH total AS
(SELECT
      C.customer_state,
      AVG(actual) AS avg_actual_Delivery_date,
      AVG(estimate) AS avg_estimate_Delivery_date

FROM `Target_SQL.customers` C INNER JOIN
       (SELECT *,
              DATE_DIFF(order_delivered_customer_date,
                     order_purchase_timestamp,day) as actual,
             DATE_DIFF(order_estimated_delivery_date,
                     order_purchase_timestamp,day) as estimate

      FROM `Target_SQL.orders`
      WHERE
              order_status = 'delivered') O ON C.customer_id = O.customer_id

GROUP BY C.customer_state)


SELECT customer_state,
      ROUND(total.avg_actual_Delivery_date -
              total.avg_estimate_Delivery_date,4) as Fast_Delivery_Time
FROM
       total

ORDER BY 2

LIMIT 5;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Fast_Delivery_Time
1	AC	-20.0875
2	RO	-19.4733
3	AP	-19.1343
4	AM	-18.9379
5	RR	-16.6585

Insights :

- The goal is to identify states where the actual delivery of orders consistently outpaces the estimated delivery times
- This involves aggregating data and computing the mean delivery time for orders.

Recommendations:

- For states with slower delivery times, conduct a detailed root cause analysis. Identify challenges, whether related to infrastructure, local regulations, or logistics partners, and work on solutions.
- Enhance customer communication for states with faster deliveries.

6.Analysis based on the payments:

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

Ans:

```
WITH Total AS
(SELECT *,
      FORMAT_DATE('%Y-%m',O.order_purchase_timestamp) AS Month_Year,

FROM `Target_SQL.orders` O INNER JOIN `Target_SQL.payments` P ON O.order_id  =  P.order_id
Where
       O.order_purchase_timestamp is not null )

SELECT
      Total.Month_Year,
      Total.payment_type,
      COUNT(Total.payment_type) AS No_of_orders

FROM Total

GROUP BY
      Total.Month_Year,
      Total.payment_type

ORDER BY 1,2;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Month_Year	payment_type	No_of_orders
1	2016-09	credit_card	3
2	2016-10	UPI	63
3	2016-10	credit_card	254
4	2016-10	debit_card	2
5	2016-10	voucher	23
6	2016-12	credit_card	1
7	2017-01	UPI	197
8	2017-01	credit_card	583
9	2017-01	debit_card	9
10	2017-01	voucher	61

Insights:

- The question groups orders by month and year (Month_Year) to provide insights into the monthly distribution of orders.

- The results are segmented by different payment types, allowing you to observe how various payment methods contribute to order placements over time.

Recommendations:

- If certain payment methods consistently have low usage, investigate the reasons behind it. Ensure that all available payment methods are user-friendly and well-promoted on your platform.
- Explore cross-promotions with payment providers to offer joint discounts or benefits to customers using specific payment methods.
- Identify the most popular payment methods each month and consider discounts, or exclusive offers for specific payment types can encourage customers to choose preferred methods.

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

Ans:

```sql
SELECT
    Payment_installments,
    COUNT(order_id) AS No_of_orders
FROM
  `Target_SQL.payments`
WHERE
    Payment_installments >=1
GROUP BY 1;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Payment_installments	No_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

Insights:

- The question groups orders based on the number of payment installments that have been paid. This provides insights into how many orders have been paid in installments and the distribution across different installment counts.
- The output show the count of orders for each installment count, indicating the popularity and usage of different installment options.

Recommendations:

- If certain installment plans are more popular, consider promoting or incentivizing those plans further. Special discounts, promotions, or loyalty points for specific installment counts can encourage customers to choose those options.
- Clearly communicate the availability and benefits of different installment plans to customers. Provide information about any associated discounts, interest rates, or flexibility in payment schedules.