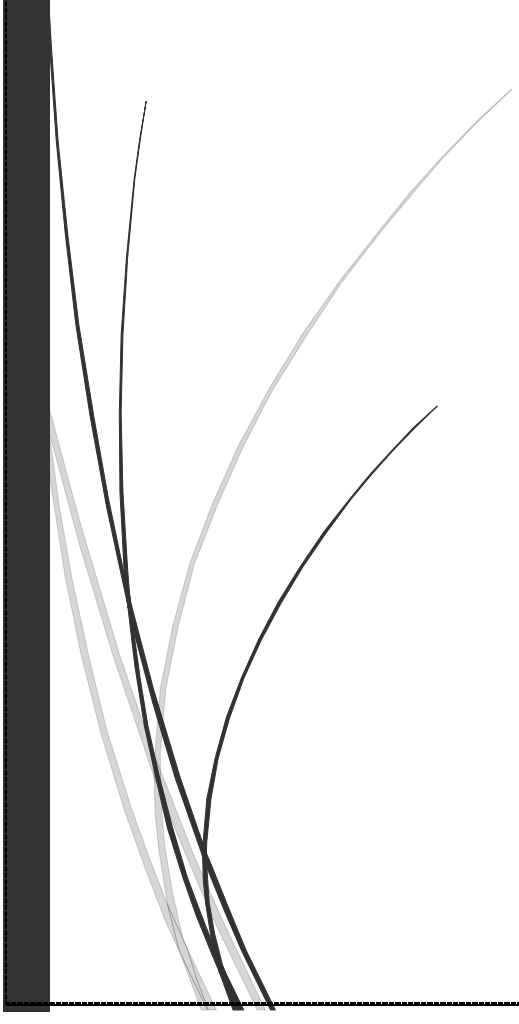
7/2/2023

Target - Biz Case Study SQL



Suguna B V

TARGET -BIZ CASE STUDY

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.

Analyzing this extensive dataset makes it 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.

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

Q 1.1) Data type of all columns in the "customers" table.

SOLUTION 1.1 →

CODE:

```
SELECT column_name,
      data_type
FROM `target.INFORMATION_SCHEMA.COLUMNS`
      WHERE TABLE_NAME = 'customers';
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | 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

Q 1.2) Get the time range between which the orders were placed.

SOLUTION 1.2 →

CODE:

```
SELECT MIN(order_purchase_timestamp) AS first_order_date,
      MAX(order_purchase_timestamp) AS last_order_date
FROM `target.orders`;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

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

Q 1.3) Count the number of Cities and States in our dataset.

SOLUTION 1.3 →

CODE:

```
SELECT COUNT(DISTINCT geolocation_city) AS city_count,
    COUNT(DISTINCT geolocation_state) AS state_count
FROM `target.geolocation`;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	city_count	state_count
1	8011	27

---

2)In-depth Exploration:

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

SOLUTION 2.1 →

CODE:

```
SELECT EXTRACT(YEAR from order_purchase_timestamp) AS year,
    EXTRACT(MONTH from order_purchase_timestamp) AS month,
    COUNT(order_id) AS order_count
FROM `target.orders`
GROUP BY year, month
ORDER BY year, month;
```

RESULT:

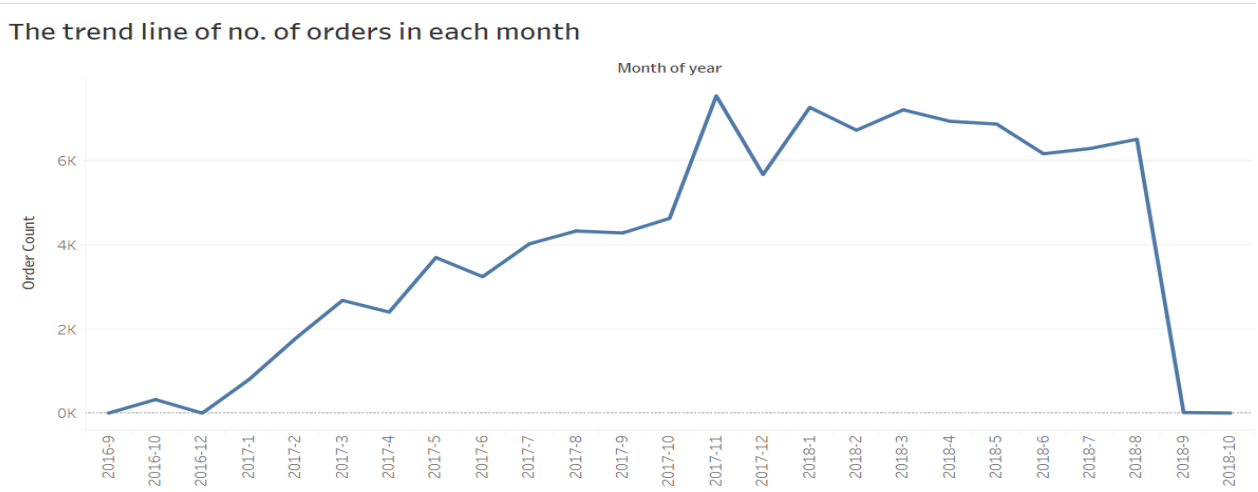
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	year	month	order_count
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

GRAPH:



INSIGHT:
To check if there is an elevation in the number of orders placed over months and years.

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

SOLUTION 2.2 →

CODE:

```
SELECT FORMAT_DATE("%B", order_purchase_timestamp) AS month,
      COUNT(order_id) AS order_count,
FROM `target.orders`
GROUP BY 1
ORDER BY 2 DESC;
```

RESULT:

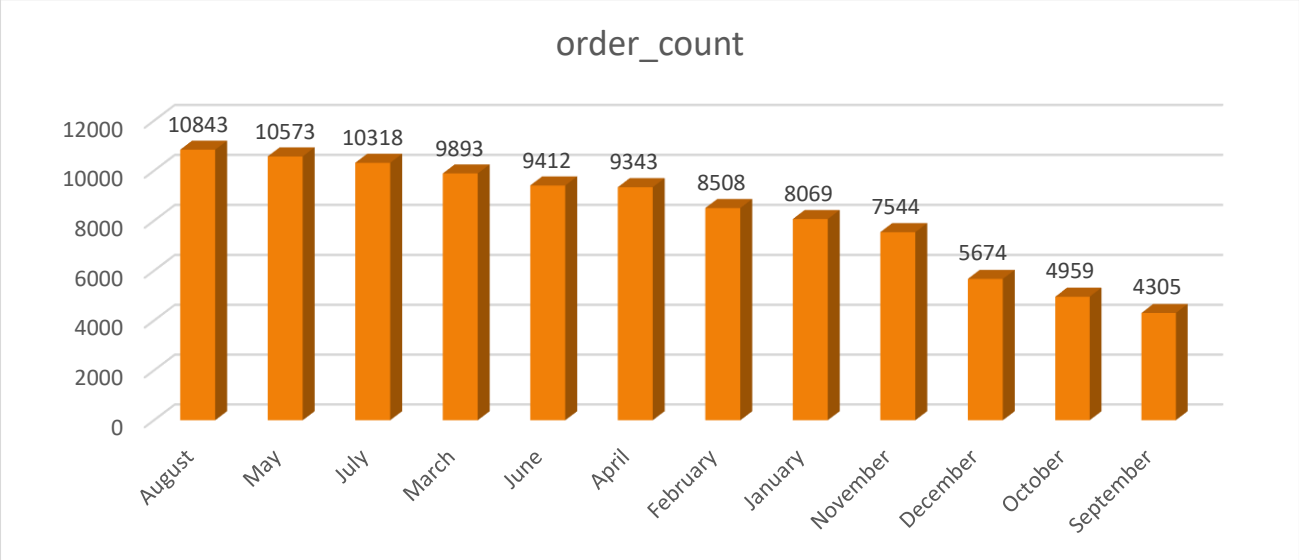
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	month	order_count
1	August	10843
2	May	10573
3	July	10318
4	March	9893
5	June	9412
6	April	9343
7	February	8508
8	January	8069
9	November	7544
10	December	5674

GRAPH:



INSIGHT:
The maximum number of orders were placed in the month of August, May and July.

RECOMMENDATIONS:

- **To keep adequate stocks in the warehouse during these months to meet its demand and supply chain.**
- **To put more staff to work during these months so that the customers are not disappointed in delivery delays.**
- **Also, the price of the products can be increased in small percentages to gain profits during peak months.**
- **To increase the orders in the rest of the months, profitable discounts are to be given on certain products.**

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

SOLUTION 2.3 →

CODE:

```
WITH t AS
(SELECT COUNT(order_id) AS order_count,
      FORMAT_TIMESTAMP("%T",order_purchase_timestamp) AS time
FROM `target.orders`
GROUP BY time)

SELECT (CASE WHEN time BETWEEN "00:00:00" AND "06:59:59" THEN "Dawn"
```

```
        WHEN time BETWEEN "07:00:00" AND "12:59:59" THEN "Mornings"
        WHEN time BETWEEN "13:00:00" AND "18:59:59" THEN "Afternoon"
        WHEN time BETWEEN "19:00:00" AND "23:59:59" THEN "Nights" END) AS time_intervals,
    SUM(order_count) AS number_of_orders
FROM t
GROUP BY 1
ORDER BY 2 DESC;
```

RESULT:

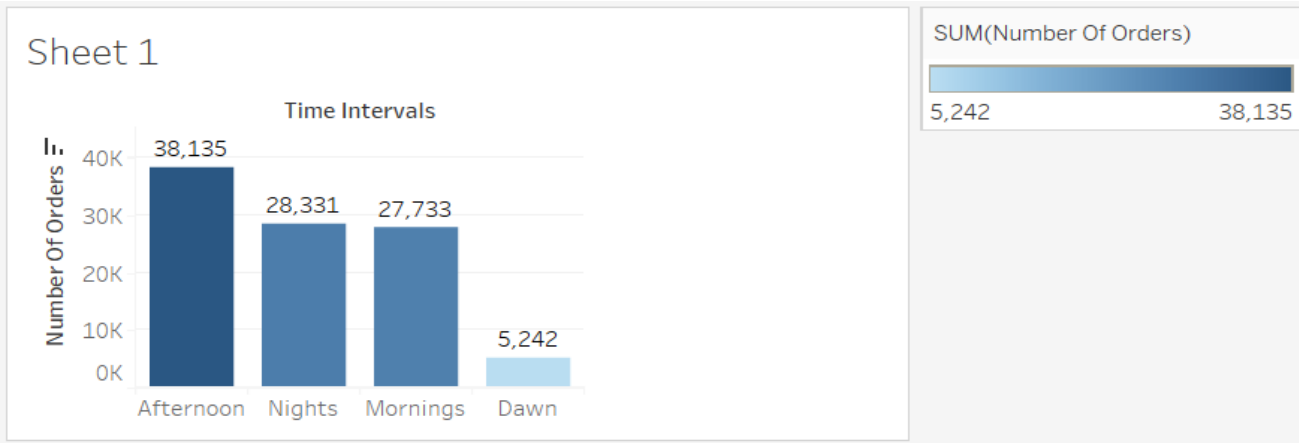
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	time_intervals	number_of_orders
1	Afternoon	38135
2	Nights	28331
3	Mornings	27733
4	Dawn	5242

GRAPH:



INSIGHT:

Most of the sales were taken place in the afternoons, moderate sales during mornings and nights, and least in the dawn.

RECOMMENDATIONS:

- **Highly recommended not to take breaks during afternoons as maximum sales were taking place during this time of the day.**
- **To increase sales during mornings and nights, offers can be given during these times of the day. E.g. Like the Late-night offers.**

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

Q 3.1) Get the month on month no. of orders placed in each state.

SOLUTION 3.1 →

CODE:

```
SELECT   c.customer_state AS state,
        FORMAT_TIMESTAMP("%m",order_purchase_timestamp) AS month,
        COUNT(order_id) AS order_count
FROM `target.orders` o

INNER JOIN `target.customers` c ON o.customer_id = c.customer_id

GROUP BY 1,2
ORDER BY 1,2;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	month	order_count
1	AC	01	8
2	AC	02	6
3	AC	03	4
4	AC	04	9
5	AC	05	10
6	AC	06	7
7	AC	07	9
8	AC	08	7
9	AC	09	5
10	AC	10	6

INSIGHT:

We can conclude that the maximum orders placed were only during certain months in each state.

RECOMMENDATIONS:

- **One can identify the seasonal sales for each state and henceforth prepare the stocks in the warehouse, and staff to supply to meet the demands of customers in individual states.**
- **The price can be raised marginally to make an increase in profits according to the seasons in each state. E.g. One can increase the price during the festive season of its respective state.**

Q 3.2) How are the customers distributed across all the states?

SOLUTION 3.2 →

CODE:

```
SELECT DISTINCT customer_state AS state,
      COUNT(customer_unique_id) OVER (PARTITION BY customer_state ORDER BY customer_state) AS
number_of_unique_customers
FROM `target.customers`
ORDER BY 2 DESC;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	number_of_unique_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

INSIGHT:
To identify the states which have the maximum and minimum number of customers in each state.

RECOMMENDATIONS:

- **To enhance focus on the states which have the least customers.**
- **Set up a marketing team to advertise the company, both online and offline modes.**
- **Rewards can be given to the stores having the highest number of customers in order to cheer up their efforts.**

---

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

Q 4.1) 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.

SOLUTION 4.1 →

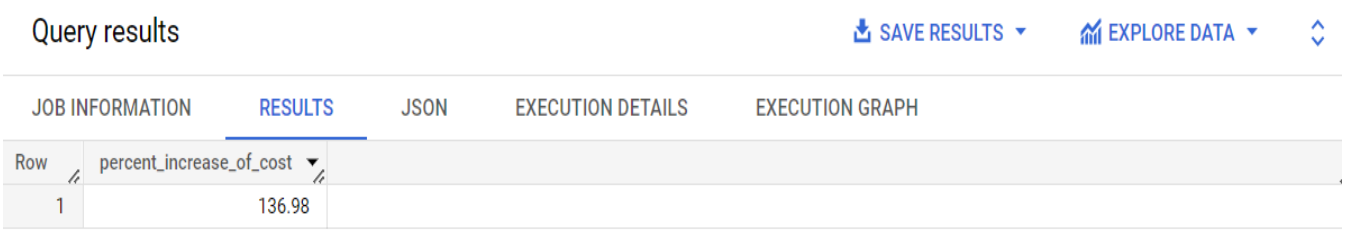
CODE:

```
WITH t AS
(SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
```

```
      ROUND(SUM(p.payment_value),2) AS cost_of_orders
FROM `target.payments` p
INNER JOIN `target.orders` o ON p.order_id = o.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND 2018
     AND EXTRACT (MONTH FROM o.order_purchase_timestamp) BETWEEN 01 AND 08
GROUP BY 1
ORDER BY 1)

SELECT *
FROM
(SELECT  ROUND((LEAD(cost_of_orders) OVER (ORDER BY year) - cost_of_orders)/cost_of_orders*100,2)
AS percent_increase_of_cost
FROM t
ORDER BY 1)
WHERE percent_increase_of_cost IS NOT NULL;
```

RESULT:



INSIGHT:

To monitor the increase in the cost of orders in percentage between the years 2017 and 2018 only for the months between January and August.

Q 4.2) Calculate the Total & Average value of order price for each state.

SOLUTION 4.2 →

CODE:

```
SELECT c.customer_state AS state,
      ROUND(SUM(oi.price),2) AS total_price,
      ROUND(AVG(oi.price),2) AS average_price

FROM `target.order_items` oi
INNER JOIN `target.orders` o ON oi.order_id = o.order_id
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 1;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	total_price	average_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
9	GO	294591.95	126.27
10	MA	119648.22	145.2

INSIGHT:

To take note of the average order price for each state compared to its total order price, so as to take action on how we can improve the average order price for the least performing states in turn to increase the profits.

Q 4.3) Calculate the Total & Average value of order freight for each state.

SOLUTION 4.3 →

CODE:

```
SELECT c.customer_state AS state,
      ROUND(SUM(oi.freight_value),2) AS total_freight,
      ROUND(AVG(oi.freight_value),2) AS average_freight

FROM `target.order_items` oi
INNER JOIN `target.orders` o ON oi.order_id = o.order_id
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 1;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	total_freight	average_freight
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
9	GO	53114.98	22.77
10	MA	31523.77	38.26

INSIGHT:

To focus on the states with greater average freight value.

RECOMMENDATION:

You can reduce the average freight value by avoiding the orders that are loss-making.

---

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

Q 5.1) 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_estimated_delivery_date - order_delivered_customer_date

SOLUTION 5.1 →

CODE:

```
SELECT order_id,
    timestamp_diff(order_delivered_customer_date, order_purchase_timestamp,    day)  AS time_to_deliver,
```

```
    timestamp_diff(order_estimated_delivery_date, order_delivered_customer_date, day) AS
diff_estimated_delivery
FROM `target.orders`
WHERE o.order_delivered_customer_date IS NOT NULL;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	time_to_deliver	diff_estimated_delivery
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

INSIGHT:

Here the negative sign in the 'diff_estimated_delivery' column represents the delay in delivery from that of the promised delivery date.

RECOMMENDATIONS:

- **To estimate accurate delivery time (in days) and gain the trust of the customer experience.**
- **Also, not to lose the customers who drop the order because of a long delivery time.**
- **To improvise the delay in delivery by using a better service.**

Q 5.2) Find out the top 5 states with the highest & lowest average freight value.

SOLUTION 5.2 →

CODE:

```
SELECT *
FROM
(SELECT state,
      average_freight,
      CONCAT("Top ",top_value) AS rank_of_states
FROM
(SELECT c.customer_state AS state,
      ROUND(AVG(oi.freight_value),2) AS average_freight,
     DENSE_RANK() OVER (ORDER BY AVG(oi.freight_value) DESC) AS top_value,

FROM `target.order_items` oi
INNER JOIN `target.orders` o ON oi.order_id = o.order_id
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
GROUP BY 1)
```

```
WHERE top_value < 6
ORDER BY top_value)

UNION ALL

(SELECT state,
       average_freight,
       CONCAT("Bottom ",bottom_value) AS rank_of_states
FROM
(SELECT c.customer_state AS state,
       ROUND(AVG(oi.freight_value),2) AS average_freight,
      DENSE_RANK() OVER (ORDER BY AVG(oi.freight_value)) AS bottom_value,

FROM `target.order_items` oi
INNER JOIN `target.orders` o ON oi.order_id = o.order_id
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
GROUP BY 1)
WHERE bottom_value < 6
ORDER BY bottom_value)

ORDER BY average_freight;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	average_freight	rank_of_states
1	SP	15.15	Bottom 1
2	PR	20.53	Bottom 2
3	MG	20.63	Bottom 3
4	RJ	20.96	Bottom 4
5	DF	21.04	Bottom 5
6	PI	39.15	Top 5
7	AC	40.07	Top 4
8	RO	41.07	Top 3
9	PB	42.72	Top 2
10	RR	42.98	Top 1

INSIGHT:

To identify the highest and lowest average freight value for each state.

RECOMMENDATION:

To draw a conclusion about the average freight values for certain states is loss-making or not. The customer might cancel the orders whose freight charges are greater than the price of the products ordered. In such cases precautions must be taken.

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

Hint: We want you to find the top 5 & the bottom 5 states arranged in increasing order of the average delivery time.

SOLUTION 5.3 →

CODE:

```
SELECT *
FROM

(SELECT state,
      average_delivery_time,
      CONCAT("Top ",top) AS rank_of_states
FROM
(SELECT state,
      ROUND(AVG(delivery_time),2) AS average_delivery_time,
      DENSE_RANK() OVER (ORDER BY AVG(delivery_time)) AS top
FROM
(SELECT c.customer_state AS state,
      timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day)  AS delivery_time
FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id)
GROUP BY 1)
WHERE top < 6)

UNION ALL

(SELECT state,
      average_delivery_time,
      CONCAT("Bottom ",bottom) AS rank_of_states
FROM
(SELECT state,
      ROUND(AVG(delivery_time),2) AS average_delivery_time,
      DENSE_RANK() OVER (ORDER BY AVG(delivery_time) DESC) AS bottom
FROM
(SELECT c.customer_state AS state,
      timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day)  AS delivery_time
FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id)
GROUP BY 1)
WHERE bottom < 6)
ORDER BY average_delivery_time;
```

RESULT:

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	average_delivery_time	rank_of_states
1	SP	8.3	Top 1
2	PR	11.53	Top 2
3	MG	11.54	Top 3
4	DF	12.51	Top 4
5	SC	14.48	Top 5
6	PA	23.32	Bottom 5
7	AL	24.04	Bottom 4
8	AM	25.99	Bottom 3
9	AP	26.73	Bottom 2
10	RR	28.98	Bottom 1

INSIGHT:

The lower the average delivery time (in days) faster the delivery of the products.

RECOMMENDATION:

To focus on the states where the average delivery time (in days) is very high, recommended actions are to be taken by the concerned services, so as to not deceive the customer expectations.

Q 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.

SOLUTION 5.4 →

CODE:

```
SELECT state,
     ROUND(AVG(diff_del_date),2) AS avg_diff_delivery_date
FROM
(SELECT c.customer_state AS state,
      timestamp_diff(order_estimated_delivery_date, order_delivered_customer_date, day) AS diff_del_date

FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
INNER JOIN `target.geolocation` g ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
WHERE o.order_delivered_customer_date IS NOT NULL)
GROUP BY 1
ORDER BY 2
LIMIT 5;
```

RESULT:

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

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

Row	state	avg_diff_delivery_date
1	AL	8.2
2	SE	8.49
3	MA	8.84
4	CE	9.72
5	ES	9.86

INSIGHT:

The top 5 states making the fastest delivery.

---

6)Analysis based on the payments:

Q 6.1) Find the month on month no. of orders placed using different payment types.

SOLUTION 6.1 →

CODE:

```
SELECT FORMAT_DATE("%Y-%m",o.order_purchase_timestamp) AS order_date,
      COUNT(CASE WHEN p.payment_type = 'credit_card' THEN p.payment_type END) AS credit_card,
      COUNT(CASE WHEN p.payment_type = 'voucher' THEN p.payment_type END) AS voucher,
      COUNT(CASE WHEN p.payment_type = 'debit_card' THEN p.payment_type END) AS debit_card,
      COUNT(CASE WHEN p.payment_type = 'UPI' THEN p.payment_type END) AS UPI,
      COUNT(CASE WHEN p.payment_type = 'not_defined' THEN p.payment_type END) AS not_defined
FROM `target.payments` p
INNER JOIN `target.orders` o ON p.order_id = o.order_id
GROUP BY 1
ORDER BY 1;
```

RESULT:

Query results

SAVE RESULTS EXPLORE DATA

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

Row	order_date	credit_card	voucher	debit_card	UPI	not_defined
1	2016-09	3	0	0	0	0
2	2016-10	254	23	2	63	0
3	2016-12	1	0	0	0	0
4	2017-01	583	61	9	197	0
5	2017-02	1356	119	13	398	0
6	2017-03	2016	200	31	590	0
7	2017-04	1846	202	27	496	0
8	2017-05	2853	289	30	772	0
9	2017-06	2463	239	27	707	0
10	2017-07	3086	364	22	845	0

INSIGHT:

To find out the total number of customers paid using different types of payment modes each month.

Q 6.2) Find the no. of orders placed on the basis of the payment installments that have been paid.

SOLUTION 6.2 →

CODE:

```
SELECT payment_installments,
      count(order_id) AS order_count
FROM `target.payments`
WHERE payment_installments >= 1
GROUP BY 1
ORDER BY 1;
```

RESULT:

Query results

SAVE RESULTS EXPLORE DATA

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

Row	payment_installments	order_count
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:

- We have 99,441 customers of data available.
- We have 96096 Unique customer IDs.
- The data given is from 04-09-2016 to 17-10-2018. A total period of about 2 years.
- The customers are from 8011 cities and 27 states.
- The number of orders placed each month gradually increased from 2016-12 till 2017-11, thereafter it was stagnant and dropped drastically in 2018-09.
- The maximum number of orders were placed during the mid-year in the months of August, May, and July and it was least during September and October.
- Most of the sales were taken place in the afternoons, the sales were moderate during mornings and nights and were least in the dawn.
- Sao Paulo state has the highest number of sellers in the country also the number of customers.
- There is an increase in the cost of orders from the year 2017 to 2018.
- we can observe the trend of increasing orders with time and also for revenue.
- Total and Average value of order price was calculated for each state.
- States AC, AL, AM had the highest average freight values.
- The state Sao Paulo has the lowest average delivery time of 8.3 days(fastest delivery of orders) and Roraima has the highest average delivery time of 28.9 days (delayed delivery of orders)

RECOMMENDATIONS :

---

1. **Customer Engagement and Retention:**
- Implement customer engagement strategies to retain and attract more customers by considering loyalty programs, personalized offers, and discounts to encourage repeat purchases.
2. **Regional Targeting:**
- Focus marketing efforts on states and cities with lower sales to boost sales in those regions.
- Consider region-specific promotions or partnerships to increase awareness and attract more customers in states that have the lowest average value of order price.
3. **Seasonal Campaigns:**
- Leverage the observation that most orders are placed during mid-year (August, May, and July).
- Plan special promotions, discounts, or exclusive products during these peak months to maximize sales.

4. **Time-Sensitive Promotions:**

- Given that most sales occur in the afternoon, consider time-sensitive promotions during peak shopping hours to boost sales.
- Analyze customer behavior to identify specific time frames for targeted marketing campaigns.

5. **Cost Management:**

- Evaluate the increase in the cost of orders and identify areas for cost optimization without compromising service quality.
- Negotiate with suppliers or explore alternative logistics solutions to reduce overall costs.

6. **Sellers and Customer Distribution:**

- Leverage the popularity of Sao Paulo by collaborating with local sellers and offering region-specific promotions.
- Explore strategies to increase the number of sellers in regions with lower sales.

7. **Delivery Time Optimization:**

- Address the issue of delayed deliveries in certain states by optimizing logistics and supply chain processes.
- Consider partnerships with local logistics providers to improve delivery times in regions with longer delivery durations.

8. **Price Optimization:**

- Regularly review and optimize product prices based on market trends, competitor pricing, and customer expectations.

9. **Inventory Management:**

- Use data to forecast demand, plan inventory, and optimize marketing campaigns for better results.

10. **Customer Feedback and Satisfaction:**

- Collect feedback from customers to understand their experience and identify areas for improvement.

*****END*****