

Targeting Success

A Case Study of Expansion into Brazil

-By Samarth Kolge

Overview of Target:

Target Corporation stands as one of the premier retail giants in the United States. Originating in 1902 as the Dayton Company, it rebranded to Target Corporation in 1962. Target distinguishes itself by offering stylish and affordable products, often collaborating with designers and brands to provide exclusive collections. With a robust online presence via its website and app, Target facilitates seamless online shopping, delivery, and in-store pickup services, catering to the evolving needs of modern consumers.

Purpose of Case Study:

In its expansion endeavors, Target has ventured into the vibrant market of Brazil, sharing invaluable insights from a dataset comprising 100,000 orders spanning 2016 to 2018. The objective of this case study is to extract meaningful insights and provide actionable feedback based on the shared dataset. Through meticulous analysis and interpretation, we aim to unveil opportunities for optimization and strategic growth, driving Target's continued success in the dynamic landscape of retail.

1. 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.

Ans- `SELECT COLUMN_NAME, DATA_TYPE
FROM `scalar-dsm1-sql-390605.target.INFORMATION_SCHEMA.COLUMNS`
WHERE TABLE_NAME = 'customers'`

The screenshot shows a SQL query editor interface. At the top, there's a toolbar with buttons for 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. A status bar on the right indicates 'Query completed.' Below the toolbar, the SQL query is entered in a text area. The query is: `SELECT COLUMN_NAME, DATA_TYPE
FROM `scalar-dsm1-sql-390605.target.INFORMATION_SCHEMA.COLUMNS`
WHERE TABLE_NAME = 'customers'`. Below the query editor, there's a section for 'Query results'. It includes tabs for 'JOB INFORMATION', 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'CHART', 'PREVIEW', and 'EXECUTION GRAPH'. The 'RESULTS' tab is active, showing a table with 5 rows and 2 columns: 'COLUMN_NAME' and 'DATA_TYPE'. The data is as follows:

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:**

Customer Data: The customer table contains unique identification numbers and location data, offering insights into demographics and geographic distribution.

- **RECOMMENDATIONS:**

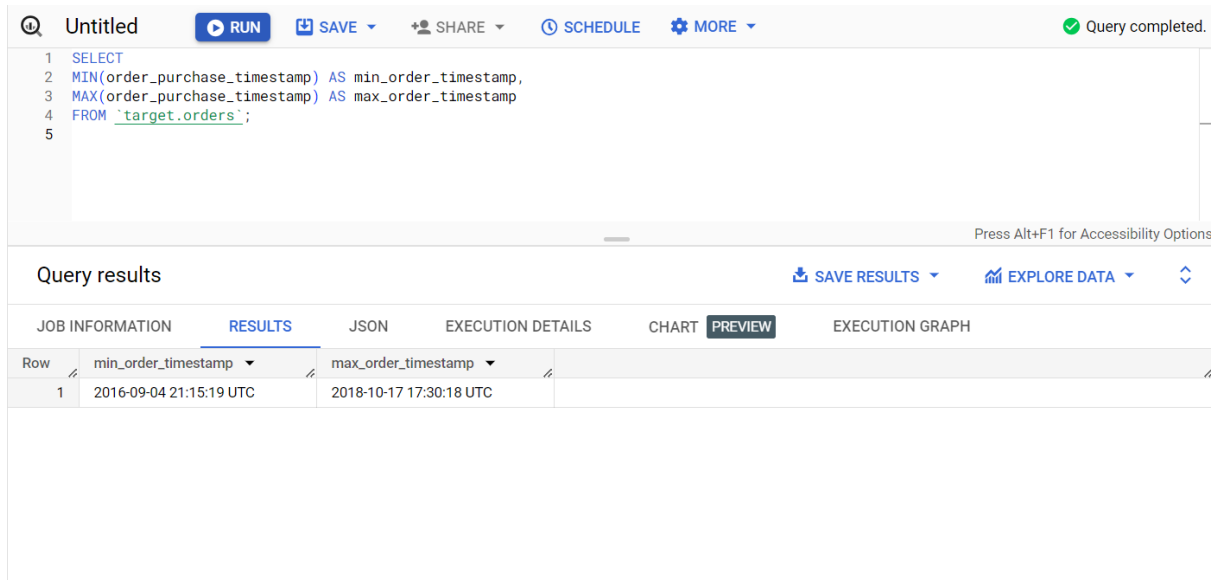
Enhanced Data Collection: Collecting additional information such as date of birth, gender, and contact details enables personalized marketing campaigns and direct communication with customers, enhancing engagement and driving sales.

- **ASSUMPTIONS :** N/A

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

Ans- `SELECT`

```
MIN(order_purchase_timestamp) AS min_order_timestamp,  
MAX(order_purchase_timestamp) AS max_order_timestamp  
FROM `target.orders`;
```



Query results

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

- **INSIGHTS :**

Data is accessible from September 2016 until October 2018, encompassing the entirety of 2017 and the initial three quarters of 2018.

- **RECOMMENDATIONS : N/A**

- **ASSUMPTIONS :**

Presuming the initial order occurred in September 2016, it's inferred that the company initiated operations in Brazil around that period.

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

Ans-

```
SELECT COUNT(DISTINCT customer_city) AS num_cities,  
COUNT(DISTINCT customer_state) AS num_states  
FROM `target.orders` o JOIN `target.customers` c  
ON o.customer_id = c.customer_id  
WHERE o.order_purchase_timestamp BETWEEN '2016-01-01' AND '2018-12-31';
```

The screenshot shows a SQL query execution interface. At the top, there's a toolbar with buttons for RUN, SAVE, SHARE, SCHEDULE, and MORE. A status bar on the right indicates "Query completed." Below the toolbar, the SQL query is displayed in a text area. The query is:

```
SELECT COUNT(DISTINCT customer_city) AS num_cities,  
COUNT(DISTINCT customer_state) AS num_states  
FROM `target.orders` o JOIN `target.customers` c  
ON o.customer_id = c.customer_id  
WHERE o.order_purchase_timestamp BETWEEN '2016-01-01' AND '2018-12-31';
```

 Below the query, there's a section titled "Query results" with tabs for JOB INFORMATION, RESULTS, JSON, EXECUTION DETAILS, CHART, PREVIEW, and EXECUTION GRAPH. The RESULTS tab is selected, showing a table with two columns: num_cities and num_states. The table has one row with values 4119 and 27 respectively. At the bottom, there are tabs for PERSONAL HISTORY and PROJECT HISTORY, and a REFRESH button.

Row	num_cities	num_states
1	4119	27

- **INSIGHTS** : This analysis tells the count of cities and states of the customers who have placed the orders.
- **RECOMMENDATIONS** : N/A
- **ASSUMPTIONS** : I am assuming that we need to consider all the states and cities even though there might not be any customers from those cities, hence choosing the table "geolocation".

2. In-depth Exploration:

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

Ans-

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

Untitled 2 ▶ RUN 📄 SAVE 👤 SHARE 🕒 SCHEDULE ⚙️ MORE ✅ Query completed.

```
4 SELECT
5 EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
6 COUNT(order_id) AS total_orders
7 FROM `target.orders`
8 GROUP BY year
9 ORDER BY year;
10
11
```

Press Alt+F1 for Accessibility Options.

Query results 📄 SAVE RESULTS 📊 EXPLORE DATA ↕

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	year	total_orders					
1	2016	329					
2	2017	45101					
3	2018	54011					

- **INSIGHTS** - Analyzing the order counts over the years can reveal there is a growing trend in the number of orders placed. A steady increase in orders over the years may suggest growth in Target's operations.
- **RECOMMENDATIONS** - Based on the analysis, There is a growing trend in the number of orders placed over the past years, Target may need to allocate resources, optimize logistics, and prepare for increased demand in the future years.
- **ASSUMPTIONS** : N/A

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

Ans-

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

Query results

Row	year	month	total_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

Results per page: 50 1 – 25 of 25

- INSIGHTS -**

We can see that May 2017, June 2017, July 2017 saw the highest sales.

- RECOMMENDATIONS -** There is evidence of monthly seasonality, Target can plan marketing campaigns, promotions, and inventory management accordingly to maximize sales during peak months.

- ASSUMPTIONS :** N/A

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

Ans-

```
SELECT
CASE WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0
AND 6 THEN 'Dawn'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND
12
THEN 'Morning'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND
18 THEN 'Afternoon'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND
23 THEN 'Night'
END AS time_of_day,
COUNT(order_id) AS total_orders,
FROM `target.orders`
GROUP BY time_of_day
ORDER BY time_of_day;
```

Query results

Row	time_of_day	total_orders
1	Afternoon	38135
2	Dawn	5242
3	Morning	27733
4	Night	28331

- INSIGHTS:**

Our analysis reveals that Brazilian customers predominantly place their orders in the afternoon, followed by the morning and night, respectively, with the fewest orders occurring in the early morning.

- RECOMMENDATIONS:**

Based on these findings, we recommend that the e-commerce platform intensify its outreach efforts during the afternoon, utilizing email alerts, promotions, notifications, and SMS messages to capitalize on peak ordering times.

- ASSUMPTIONS:** N/A

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

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

Ans-

```
SELECT CONCAT(EXTRACT(YEAR FROM order_purchase_timestamp), '-',
EXTRACT(MONTH FROM order_purchase_timestamp)) AS year_month,
c.customer_state, COUNT(*) AS order_numbers
FROM `target.orders` o JOIN `target.customers` c
ON o.customer_id = c.customer_id
GROUP BY year_month, c.customer_state
ORDER BY year_month, c.customer_state;
```

Untitled ▶ RUN 📄 SAVE 👤 SHARE 🕒 SCHEDULE 🟢 Query completed.

```
1 SELECT
2 CONCAT(EXTRACT(YEAR FROM order_purchase_timestamp), '-', EXTRACT(MONTH FROM order_purchase_timestamp))
   AS year_month, c.customer_state, COUNT(*) AS order_numbers
```

Press Alt+F1 for Accessibility Options.

Query results 📄 SAVE RESULTS 📊 EXPLORE DATA

Row	year_month	customer_state	order_numbers
1	2016-10	AL	2
2	2016-10	BA	4
3	2016-10	CE	8
4	2016-10	DF	6
5	2016-10	ES	4
6	2016-10	GO	9
7	2016-10	MA	4
8	2016-10	MG	40
9	2016-10	MT	3
10	2016-10	PA	4

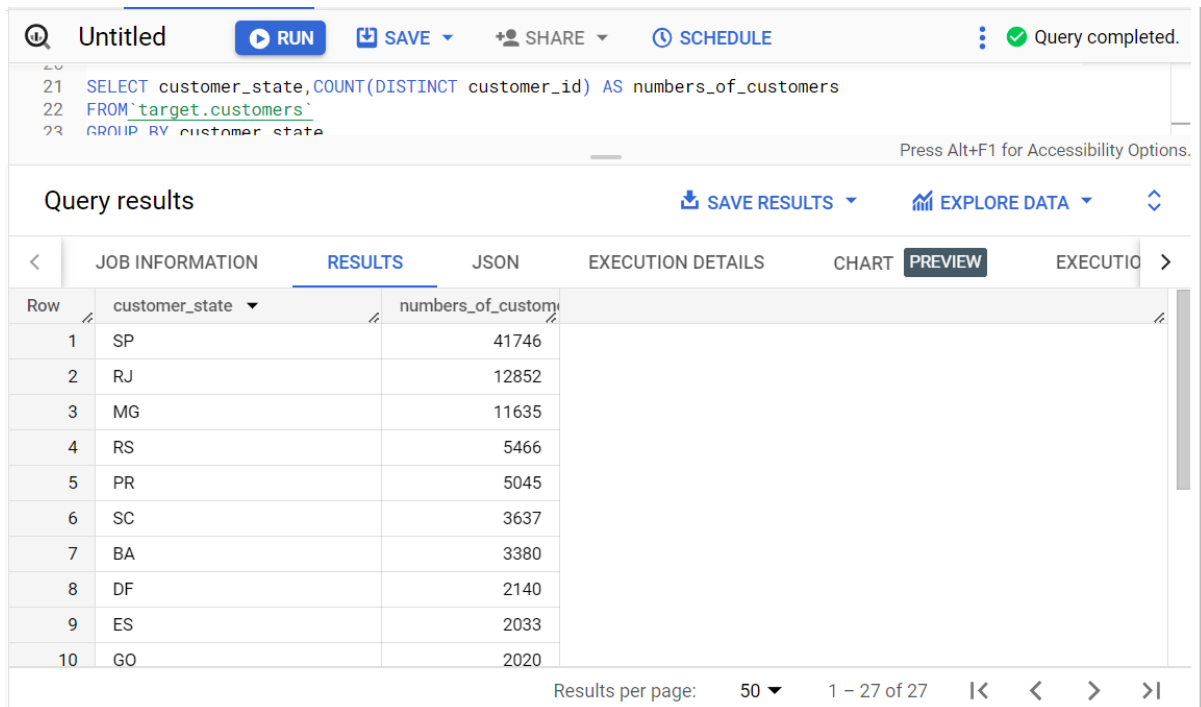
Results per page: 50 1 – 50 of 565

- **INSIGHTS** - N/A
- **RECOMMENDATIONS** – N/A
- **ASSUMPTIONS:** N/A

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

Ans-

```
SELECT customer_state,
COUNT(DISTINCT customer_id) AS numbers_of_customers
FROM `target.customers`
GROUP BY customer_state
ORDER BY numbers_of_customers DESC;
```



The screenshot shows a SQL query interface with a query editor at the top and a results table below. The query is:

```
SELECT customer_state, COUNT(DISTINCT customer_id) AS numbers_of_customers
FROM `target.customers`
GROUP BY customer_state
ORDER BY numbers_of_customers DESC;
```

 The results table has two columns: 'customer_state' and 'numbers_of_customers'. It lists 10 rows of data, sorted by the number of customers in descending order. The states and their respective customer counts are: SP (41746), RJ (12852), MG (11635), RS (5466), PR (5045), SC (3637), BA (3380), DF (2140), ES (2033), and GO (2020). The interface includes buttons for 'RUN', 'SAVE', 'SHARE', and 'SCHEDULE' at the top, and tabs for 'JOB INFORMATION', 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'CHART', 'PREVIEW', and 'EXECUTION' below the query editor. The 'RESULTS' tab is currently selected.

Row	customer_state	numbers_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:**

Our analysis of customer distribution across Brazilian states reveals that São Paulo (SP), Rio de Janeiro (RJ), and Minas Gerais (MG) are the top states with a significant customer base, while Acre (AC), Amapá (AP), and Roraima (RR) have fewer clients.

- **RECOMMENDATIONS:**

To address states with fewer customers, we suggest conducting targeted market analysis and promotions at the state level to increase brand awareness and attract new clientele.

- **ASSUMPTIONS:** N/A

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).

Ans-

```
SELECT
ROUND((SUM(CASE WHEN EXTRACT(YEAR FROM
o.order_purchase_timestamp ) = 2018 AND EXTRACT(MONTH FROM
o.order_purchase_timestamp) <= 8 THEN p.payment_value ELSE 0
END) -
SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) =
2017 AND EXTRACT(MONTH FROM o.order_purchase_timestamp) <= 8
THEN p.payment_value ELSE 0 END)) /
SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) =
2017 AND EXTRACT(MONTH FROM o.order_purchase_timestamp) <= 8
THEN p.payment_value ELSE 1 END) * 100, 2) AS
percentage_increase
FROM `target.payments` AS p join `target.orders` as o
ON p.order_id = o.order_id
```

The screenshot shows a SQL query execution interface. At the top, there's a toolbar with buttons for 'RUN', 'SAVE', 'SHARE', and 'SCHEDULE'. Below the toolbar, the query is displayed in a text area. The query calculates the percentage increase in the cost of orders from 2017 to 2018, considering only orders from January to August. The query is as follows:

```
SELECT
ROUND((SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp ) = 2018 AND EXTRACT(MONTH FROM o.
order_purchase_timestamp) <= 8 THEN p.payment_value ELSE 0 END) -
SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM o.
order_purchase_timestamp) <= 8 THEN p.payment_value ELSE 0 END)) /
SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM o.
order_purchase_timestamp) <= 8 THEN p.payment_value ELSE 1 END) * 100, 2) AS percentage_increase
FROM `target.payments` AS p join `target.orders` as o
ON p.order_id = o.order_id
```

Below the query, the 'Query results' section is visible. It shows a table with one row and one column, 'percentage_increase', with a value of 134.07.

Row	percentage_increase
1	134.07

- INSIGHTS:**
 The cost of orders has surged by approximately 134% from 2017 to 2018, representing a substantial increase, more than doubling within this period.
- RECOMMENDATIONS:**
 The remarkable growth observed warrants a thorough analysis of the business decisions and strategies implemented during this period. Lessons learned can be leveraged to inform and optimize future business practices.
- ASSUMPTIONS:**
 N/A

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

Ans-

```
SELECT customer_state,
ROUND(SUM(p.payment_value),2) AS total_order_price,
ROUND(AVG(p.payment_value),2) AS average_order_price
FROM `target.payments` AS p JOIN `target.orders` AS o
ON p.order_id = o.order_id
JOIN `target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY customer_state;
```

Query results

Row	customer_state	total_order_price	average_order_price
1	RJ	2144379.69	158.53
2	RS	890898.54	157.18
3	SP	5998226.96	137.5
4	DF	355141.08	161.13
5	PR	811156.38	154.15
6	MT	187029.29	195.23
7	MA	152523.02	198.86
8	AL	96962.06	227.08
9	MG	1872257.26	154.71
10	PE	324850.44	187.99

Results per page: 50 1 – 27 of 27

- INSIGHTS:**
 - The query result provides valuable insights for businesses, offering a deeper understanding of customer shopping behavior and potential revenue fluctuations.
 - A low average order value may indicate that customers are purchasing only one or a few products per order, or products with lower prices, but with higher frequency.
 - Conversely, a high average order value may suggest a customer base that orders less frequently but purchases more products or higher-priced items.
- RECOMMENDATIONS:**
 - Utilize these insights to segment the customer base into various groups, such as small vs. big spenders and low vs. high-frequency shoppers, based on their purchase history.
 - Leverage this metric to inform decisions regarding pricing strategies, product assortment, and promotional activities.
- ASSUMPTIONS:** The order price for each state is assumed to be the price of each order, excluding freight costs.

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

Ans-

```
SELECT customer_state,
SUM(i.freight_value) AS total_freight_value,
AVG(i.freight_value) AS average_freight_value
FROM `target.order_items` AS i JOIN `target.orders` AS o
ON i.order_id = o.order_id
JOIN `target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY customer_state;
```

Query results

Row	customer_state	total_freight_value	average_freight_value
1	SP	718723.0699999...	15.14727539041...
2	RJ	305589.3100000...	20.96092393168...
3	PR	117851.6800000...	20.53165156794...
4	SC	89660.26000000...	21.47036877394...
5	DF	50625.49999999...	21.04135494596...
6	MG	270853.4600000...	20.63016680630...
7	PA	38699.30000000...	35.83268518518...
8	BA	100156.6799999...	26.36395893656...
9	GO	53114.97999999...	22.76681525932...
10	RS	135522.7400000...	21.73580433039...

Results per page: 50 1 - 27 of 27

- INSIGHTS:**

By analyzing the data provided, we can determine state rankings based on the average freight value, representing the transportation costs associated with a single purchase from each state.

- RECOMMENDATIONS:**

- States with high average freight values should undergo freight auditing and cost reduction studies to optimize transportation expenses.
- Implement strategic adjustments such as increasing lead times for deliveries, shipping on non-peak days, consolidating smaller shipments, shipping products less frequently but in larger quantities, and reducing pallet space.
- Focus efforts on bringing merchants of frequently purchased goods closer to the same zone or plan ahead to preserve storage space for these goods.

- ASSUMPTIONS:** N/A

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

- 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.

Ans-

```
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`;
```

🔍

Untitled

▶

RUN

⚙️

MORE

▼

💾

SAVE

▼

👤

SHARE

▼

🕒

SCHEDULE

✔️

Query completed.

67

SELECT order_id,

68

TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_to_deliver,

69

TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estimated_delivery

Press Alt+F1 for Accessibility Options.

Query results

📄

SAVE RESULTS

▼

📊

EXPLORE DATA

▼

⬆️

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

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

Results per page:

50

1 – 50 of 99441

⏪

⏩

⏴

⏵

- INSIGHTS - Delivery Time:** This provides an understanding of the average time taken to deliver orders to customers after the purchase date. A shorter delivery time indicates better shipping efficiency.

Difference in Estimated & Actual Delivery: This helps identify how well the estimated delivery dates align with the actual delivery dates. A negative value indicates orders were delivered earlier than expected, while a positive value indicates delays.

- RECOMMENDATIONS** - Analyzing delivery times and the differences between estimated and actual delivery dates can help identify areas for improvement in Target's logistics and delivery processes. If the "diff_estimated_delivery" is consistently negative, it indicates that Target is delivering orders before the estimated delivery date, which can enhance customer satisfaction.

- ASSUMPTIONS:** N/A

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

Ans-

```
WITH StateFreight AS (
  SELECT customer_state, AVG(freight_value) AS avg_freight
  FROM `target.order_items` AS i JOIN `target.orders` AS o
  ON i.order_id = o.order_id
  JOIN `target.customers` AS c
  ON o.customer_id = c.customer_id
  GROUP BY customer_state
)
(SELECT customer_state, avg_freight
FROM StateFreight
ORDER BY avg_freight DESC
LIMIT 5)
UNION ALL
(SELECT customer_state, avg_freight
FROM StateFreight
ORDER BY avg_freight ASC
LIMIT 5);
```

Query results

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

- INSIGHTS:**

- States with the highest freight value indicate that customers pay freight charges for each order on average.
- Conversely, states with lower freight value suggest that customers pay a standard, lower freight cost for each order on average.

- RECOMMENDATIONS:**

- Conduct freight auditing and cost reduction studies for states with high average freight values to optimize transportation expenses.

- Implement strategic adjustments such as increasing lead times for deliveries, shipping on non-peak days, consolidating smaller shipments, shipping products less frequently but in larger quantities, and reducing pallet space.
 - Focus efforts on bringing merchants of frequently purchased goods closer to the same zone or plan ahead to preserve storage space for these goods.
-
- **ASSUMPTIONS:** N/A

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

Ans-

```
WITH DeliveryTimeDays AS (
  SELECT customer_state,
  AVG(TIMESTAMP_DIFF(order_delivered_customer_date,
  order_purchase_timestamp, DAY)) AS avg_delivery_time_days
  FROM `target.orders` o JOIN `target.customers` AS c
  ON o.customer_id = c.customer_id
  GROUP BY customer_state )
(SELECT customer_state, avg_delivery_time_days
FROM DeliveryTimeDays
ORDER BY avg_delivery_time_days DESC
LIMIT 5)
UNION ALL
(SELECT customer_state, avg_delivery_time_days
FROM DeliveryTimeDays
ORDER BY avg_delivery_time_days ASC
LIMIT 5);
```

Untitled ▶ RUN ⚙️ MORE 💾 SAVE 👤 SHARE 🕒 SCHEDULE ✅ Query completed.

```
88 WITH DeliveryTimeDays AS (
89 SELECT customer_state,
90 AVG(TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS avg_delivery_time_days
91 FROM `target.orders` o JOIN `target.customers` AS c
```

Press Alt+F1 for Accessibility Options

Query results 📄 SAVE RESULTS 📊 EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	avg_delivery_time_days					
1	RR	28.97560975609...					
2	AP	26.73134328358...					
3	AM	25.98620689655...					
4	AL	24.04030226700...					
5	PA	23.31606765327...					
6	SP	8.298061489072...					
7	PR	11.52671135486...					
8	MG	11.54381329810...					
9	DF	12.50913461538...					
10	SC	14.47956019171...					

- **INSIGHTS:**

- States with the longest delivery times indicate that, on average, customers experience lengthy waiting periods to receive their orders.
- Conversely, states with the shortest delivery times suggest that customers prioritize prompt delivery of their requested goods.

- **RECOMMENDATIONS:**

- Analyze and address flaws in the logistics operation for states with the longest delivery times to enhance the overall customer experience.

- **ASSUMPTIONS:** N/A

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.

Ans-

```
WITH FDelivery AS (
  SELECT customer_state,
  AVG(TIMESTAMP_DIFF(order_delivered_customer_date,
  order_purchase_timestamp, DAY)) AS avg_delivery_time_days,
  AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,
  order_delivered_customer_date, DAY)) AS
  avg_diff_estimated_delivery_days
  FROM `target.orders` o JOIN `target.customers` AS c
  ON o.customer_id = c.customer_id
  GROUP BY customer_state )
  SELECT customer_state,
  AVG(avg_delivery_time_days - avg_diff_estimated_delivery_days)
  AS avg_fast_delivery_days
  FROM FDelivery
  GROUP BY customer_state
  ORDER BY avg_fast_delivery_days DESC
  LIMIT 5;
```

Query completed.

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	avg_fast_delivery_da					
1	AL	16.09319899244...					
2	RR	12.56097560975...					
3	MA	12.34867503486...					
4	SE	11.85671641791...					
5	CE	10.86004691164...					

- **INSIGHTS:** These top 5 states experience exceptionally fast delivery times compared to the estimated date of delivery, providing customers with early receipt of their ordered products.
- **RECOMMENDATIONS:** Analyze the logistics operation strategies and methodologies employed by these states to achieve fast delivery times and consider implementing similar practices in other states to improve overall delivery efficiency.
- **ASSUMPTIONS:** N/A

6. Analysis based on the payments:

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

Ans-

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
payment_type,
COUNT(DISTINCT o.order_id) AS num_orders
FROM `target.orders` o join `target.payments` p
ON o.order_id = p.order_id
GROUP BY year, month, payment_type
ORDER BY year, month, payment_type;
```

Untitled ▶ RUN ⚙ MORE 💾 SAVE 👤 SHARE 🕒 SCHEDULE ✅ Query completed.

```
121 SELECT
122 EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
123 EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
```

Press Alt+F1 for Accessibility Options

Query results 📄 SAVE RESULTS 📊 EXPLORE DATA ⌵

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	year	month	payment_type	num_orders		
1	2016	9	credit_card	3		
2	2016	10	UPI	63		
3	2016	10	credit_card	253		
4	2016	10	debit_card	2		
5	2016	10	voucher	11		
6	2016	12	credit_card	1		
7	2017	1	UPI	197		
8	2017	1	credit_card	582		
9	2017	1	debit_card	9		
10	2017	1	voucher	33		

Results per page: 50 1 - 50 of 90 |< < > >|

- **INSIGHTS:**

- The analysis indicates that customers are more inclined to use credit cards for payment compared to debit cards, UPI, or vouchers.
- Payments without proper payment types, totalling 3, can be disregarded in the analysis.

- **RECOMMENDATIONS:**

- Considering Brazilian customers' preference for paying with credit cards, offering credit-related incentives in partnership with banks may lead to an uptick in customer orders.

- **ASSUMPTIONS:**

- N/A

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

Ans-

```
SELECT payment_installments,
COUNT(DISTINCT order_id) AS num_orders_placed
FROM `target.payments`
GROUP BY payment_installments
ORDER BY payment_installments;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS CHART PREVIEW EXECUTION GRAPH

Row	payment_installment	num_orders_placed
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644

Results per page: 50 1 - 24 of 24

- **INSIGHTS:** Approximately 50% of the orders placed involve at least one installment payment.
- **RECOMMENDATIONS:** N/A
- **ASSUMPTIONS:** N/A