Target Corporation

Naynish Ladkat

Overview

Target is the eighth-largest retailer in the United States, founded in 1902. It operates as a general merchandise retailer, offering various products including clothing, household essentials, electronics, toys, beauty products, groceries, and more.

Goals

Perform analysis of the Target Company data to provide and drive insights that would help the company make strategic decisions.

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

1. Data type of columns in a table

Ans -

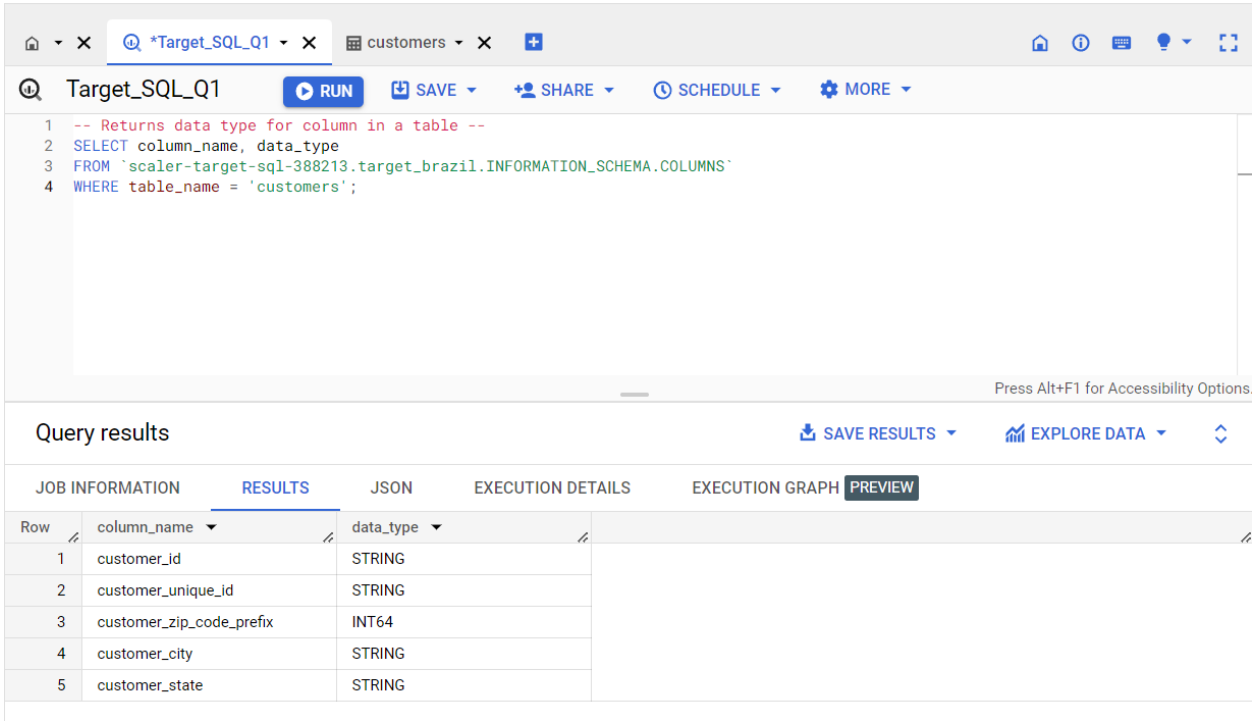
Query -

```
-- Returns data type for a column in a table --

SELECT column_name, data_type

FROM `scaler-target-sql-388213.target_brazil.INFORMATION_SCHEMA.COLUMNS`

WHERE table_name = 'customers';
```



GEOLOCATION

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

Row	column_name	data_type
1	geolocation_zip_code_prefix	INT64
2	geolocation_lat	FLOAT64
3	geolocation_lng	FLOAT64
4	geolocation_city	STRING
5	geolocation_state	STRING

ORDER ITEMS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64

ORDER REVIEWS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	review_id	STRING
2	order_id	STRING
3	review_score	INT64
4	review_comment_title	STRING
5	review_creation_date	TIMESTAMP
6	review_answer_timestamp	TIMESTAMP

ORDERS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

PAYMENTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	order_id	STRING
2	payment_sequential	INT64
3	payment_type	STRING
4	payment_installments	INT64
5	payment_value	FLOAT64

PRODUCTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	product_id	STRING
2	product_category	STRING
3	product_name_length	INT64
4	product_description_length	INT64
5	product_photos_qty	INT64
6	product_weight_g	INT64
7	product_length_cm	INT64
8	product_height_cm	INT64
9	product_width_cm	INT64

SELLERS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	seller_id	STRING
2	seller_zip_code_prefix	INT64
3	seller_city	STRING
4	seller_state	STRING

2. Time period for which the data is given

4th Sept 2016 to 17th October 2018

Query - For the End date

```
SELECT max(order_purchase_timestamp)
FROM `target_brazil.orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	f0_
1	2018-10-17 17:30:18 UTC

Query - For the Start date

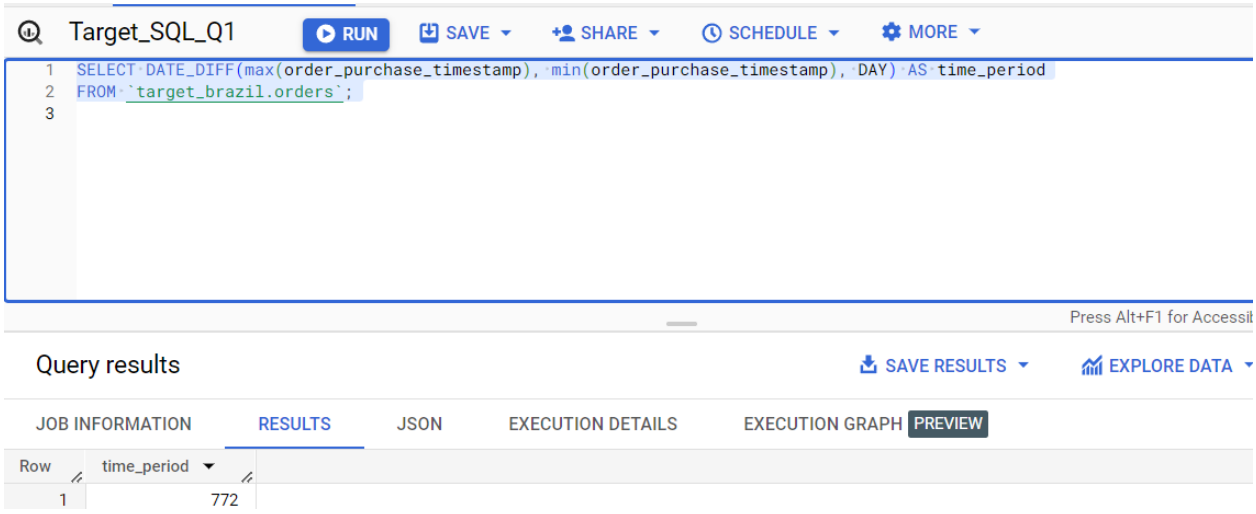
Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	f0_
1	2016-09-04 21:15:19 UTC

ANS - In Days -

Query -
```
SELECT DATE_DIFF(max(order_purchase_timestamp), min(order_purchase_timestamp), DAY) AS time_period
FROM `target_brazil.orders`;
```



3. Cities and States of customers ordered during the given period

Query -

```
SELECT DISTINCT customer_city,customer_state
FROM `target_brazil.customers` as C INNER JOIN `target_brazil.orders` as O ON
C.customer_id = O.customer_id;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT

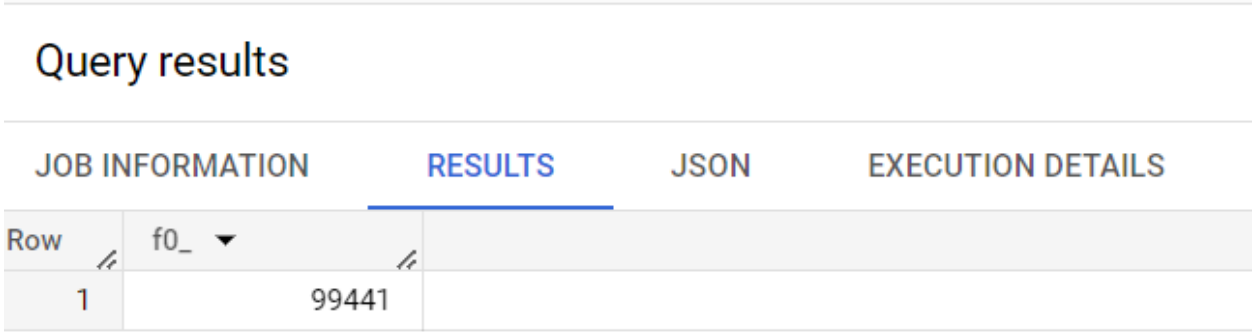
In-depth Exploration:

1. Is there a growing trend in e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Query -

```
SELECT count(order_id)
FROM `target_brazil.orders`
```

The total number of orders is 99441 between the time period



2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query -

```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 0 AND EXTRACT(HOUR FROM order_purchase_timestamp) < 12 THEN 'Morning'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 12 AND EXTRACT(HOUR FROM order_purchase_timestamp) < 18 THEN 'Afternoon'
    ELSE 'Night'
  END AS purchase_period,
  COUNT(*) AS purchase_count
FROM
  `target_brazil.orders`
GROUP BY
  purchase_period;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	purchase_period	purchase_count
1	Morning	26980
2	Afternoon	38361
3	Night	34100

Evolution of E-commerce orders in the Brazil region:

1. Get month on month orders by states

Query -

```
SELECT O.order_id , O.order_purchase_timestamp, L.customer_state
FROM `target_brazil.orders` AS O
INNER JOIN `target_brazil.customers` AS L ON O.customer_id = L.customer_id;
```

Query results

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

Row	order_id	order_purchase_timestamp	customer_state
1	6190a94657e1012983a274b8...	2017-07-11 13:36:30 UTC	AL
2	52cb9b4d5ee3ce7d1e2a8d9c2...	2018-07-11 20:24:49 UTC	SE
3	274a7f7e4f1c17b7434a830e9...	2018-06-23 13:25:15 UTC	SE
4	d430c6c36d198f044555a51a5...	2017-07-30 16:45:22 UTC	AL
5	48a310c40917683b0b399849...	2017-02-24 13:50:32 UTC	PI

2. Distribution of customers across the states in Brazil

Query -

```
SELECT customer_state, COUNT(*) AS customer_count
FROM `target_brazil.customers`
GROUP BY customer_state;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	customer_count
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747

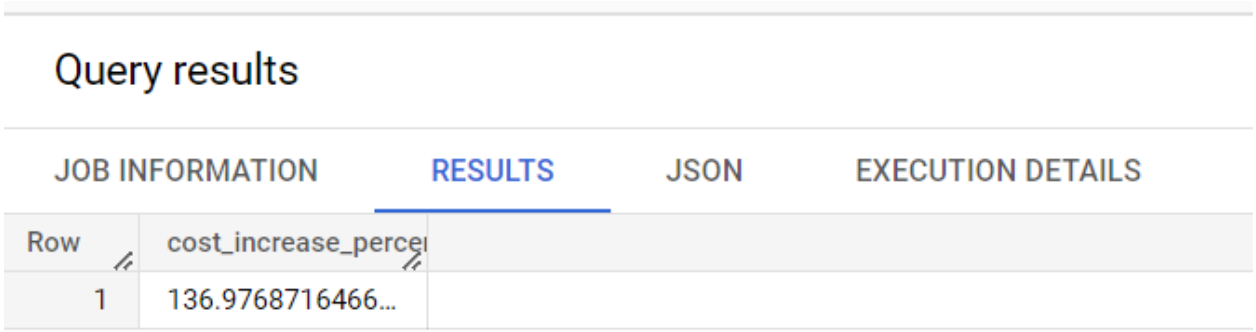
Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table

Query -

```
SELECT
  (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) -
  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)) /
```

```
  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) *
100 AS cost_increase_percentage
FROM
  `target_brazil.payments` AS p
JOIN
  `target_brazil.orders` AS o ON p.order_id = o.order_id
WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018);
```



2. Mean & Sum of price and freight value by customer state

Query -

```
SELECT
  c.customer_state,
  AVG(oi.price) AS average_price,
  SUM(oi.price) AS total_price,
  AVG(oi.freight_value) AS average_freight,
  SUM(oi.freight_value) AS total_freight
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.order_items` AS oi ON o.order_id = oi.order_id
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state;
```

Query results

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

Row	customer_state	average_price	total_price	average_freight	total_freight
1	MT	148.2971848341...	156453.5299999...	28.16628436018...	29715.43000000...
2	MA	145.2041504854...	119648.2199999...	38.25700242718...	31523.77000000...
3	AL	180.8892117117...	80314.81	35.84367117117...	15914.58999999...
4	SP	109.6536291597...	5202955.050001...	15.14727539041...	718723.0699999...
5	MG	120.7485741488...	1585308.029999...	20.63016680630...	270853.4600000...
6	PE	145.5083222591...	262788.0299999...	32.91786267995...	59449.65999999...
7	RJ	125.1178180945...	1824092.669999...	20.96092393168...	305589.3100000...
8	DF	125.7705486284...	302603.9399999...	21.04135494596...	50625.49999999...
9	RS	120.3374530874...	750304.0200000...	21.73580433039...	135522.7400000...
10	SE	153.0411688311...	58920.85000000...	36.65316883116...	14111.46999999...

5. Analysis on sales, freight and delivery time

Calculate days between purchasing, delivering and estimated delivery

Query -

```
SELECT
  order_purchase_timestamp,
  order_delivered_customer_date,
  order_estimated_delivery_date,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
days_to_delivery,
  DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS
days_to_estimated_delivery
FROM
  `target_brazil.orders`;
```

Query results

SAVE RESULTS

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

Row	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	days_to_delivery	days_to_estimated_d
1	2017-12-09 10:16:45 UTC	null	2018-01-29 00:00:00 UTC	null	50
2	2018-08-10 15:14:50 UTC	null	2018-08-17 00:00:00 UTC	null	6
3	2017-05-13 21:23:34 UTC	null	2017-06-27 00:00:00 UTC	null	44
4	2016-10-07 19:17:00 UTC	null	2016-12-01 00:00:00 UTC	null	54
5	2016-10-05 01:47:40 UTC	null	2016-12-01 00:00:00 UTC	null	56
6	2016-10-07 22:45:28 UTC	null	2016-12-01 00:00:00 UTC	null	54
7	2016-10-05 16:57:30 UTC	null	2016-12-01 00:00:00 UTC	null	56
8	2018-03-08 07:06:35 UTC	null	2018-04-19 00:00:00 UTC	null	41
9	2018-08-05 07:21:56 UTC	null	2018-08-09 00:00:00 UTC	null	3
10	2018-08-05 17:00:00 UTC	null	2018-08-09 00:00:00 UTC	null	3

Results per page: 50 1 – 50

2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- time_to_delivery = order_delivered_customer_date-order_purchase_timestamp

Query -

```
SELECT

  order_purchase_timestamp,

  order_delivered_customer_date,

  TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, HOUR) AS
time_to_delivery_hours,

  TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, MINUTE) AS
time_to_delivery_minutes

FROM

  `target_brazil.orders`;
```

Query results

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

Row	order_purchase_timestamp	order_delivered_customer_date	time_to_delivery_hou	time_to_delivery_min
1	2017-11-25 11:10:33 UTC	null	null	null
2	2017-12-05 01:07:58 UTC	null	null	null
3	2017-12-05 01:07:52 UTC	null	null	null
4	2018-02-09 17:21:04 UTC	null	null	null
5	2017-11-06 13:12:34 UTC	null	null	null
6	2017-04-20 12:45:34 UTC	null	null	null
7	2017-07-13 11:03:05 UTC	null	null	null
8	2017-07-11 13:36:30 UTC	null	null	null
9	2017-07-29 18:05:07 UTC	null	null	null
10	2017-07-13 10:02:47 UTC	null	null	null
11	2017-07-19 12:44:59 UTC	null	null	null

Results per page: 5

- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Query -

```
SELECT
  order_estimated_delivery_date,
  order_delivered_customer_date,
  TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, HOUR)
AS diff_estimated_delivery_hours,
  TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, MINUTE)
AS diff_estimated_delivery_minutes
FROM
  `target_brazil.orders`;
```

Query results

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

Row	order_purchase_timestamp	order_delivered_customer_date	time_to_delivery_hou	time_to_delivery_min
1	2017-11-25 11:10:33 UTC	null	null	null
2	2017-12-05 01:07:58 UTC	null	null	null
3	2017-12-05 01:07:52 UTC	null	null	null
4	2018-02-09 17:21:04 UTC	null	null	null
5	2017-11-06 13:12:34 UTC	null	null	null
6	2017-04-20 12:45:34 UTC	null	null	null
7	2017-07-13 11:03:05 UTC	null	null	null
8	2017-07-11 13:36:30 UTC	null	null	null
9	2017-07-29 18:05:07 UTC	null	null	null
10	2017-07-13 10:02:47 UTC	null	null	null
11	2017-07-19 12:44:59 UTC	null	null	null

Results per page: 5

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Query -

```
SELECT
  c.customer_state,
  AVG(oi.freight_value) AS average_freight_value,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
HOUR)) AS average_time_to_delivery_hours,
  AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date,
HOUR)) AS average_diff_estimated_delivery_hours
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.order_items` AS oi ON o.order_id = oi.order_id
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state;
```

Query results

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

Row	customer_state	average_freight_valu	average_time_to_deli	average_diff_estimat
1	SP	15.14727539041...	208.8689145834...	251.8935684602...
2	RJ	20.96092393168...	363.0629860031...	271.0438993355...
3	PR	20.53165156794...	286.2403965303...	306.5740839086...
4	SC	21.47036877394...	359.5256222547...	260.5500244021...
5	DF	21.04135494596...	310.5184713375...	275.4195329087...
6	MG	20.63016680630...	287.1123325849...	302.9130603081...
7	PA	35.83268518518...	569.6005692599...	325.2893738140...
8	BA	26.36395893656...	461.4512625576...	246.5734455606...

4. Sort the data to get the following:

5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Query -

```
SELECT
  c.customer_state,
  AVG(oi.freight_value) AS average_freight_value
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.order_items` AS oi ON o.order_id = oi.order_id
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  average_freight_value DESC
LIMIT
  5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	average_freight_valu
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

Query -

```
SELECT
  c.customer_state,
  AVG(oi.freight_value) AS average_freight_value
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.order_items` AS oi ON o.order_id = oi.order_id
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  average_freight_value ASC
LIMIT
  5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	average_freight_valu
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

6. Top 5 states with highest/lowest average time to delivery

Query -

```
SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
HOUR)) AS average_time_to_delivery_hours
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  average_time_to_delivery_hours DESC
LIMIT
  5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	average_time_to_deli
1	RR	704.7317073170...
2	AP	651.9701492537...
3	AM	633.6965517241...
4	AL	588.5415617128...
5	PA	570.0507399577...

Query -

```
SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
HOUR)) AS average_time_to_delivery_hours
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  average_time_to_delivery_hours ASC
LIMIT
  5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	average_time_to_deli
1	SP	209.7709840721...
2	PR	287.3026609790...
3	MG	287.7454865697...
4	DF	310.7235576923...
5	SC	358.5167747392...

7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Query -

```
SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date,
DAY)) AS average_delivery_difference_days
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
HAVING
  average_delivery_difference_days > 0
ORDER BY
  average_delivery_difference_days DESC
LIMIT
  5;
```

ANS - No Data

6. Payment type analysis:

1. Month over Month count of orders for different payment types

```
SELECT
  DATE_TRUNC(o.order_purchase_timestamp, MONTH) AS order_month,
  p.payment_type,
  COUNT(*) AS order_count
FROM
  `target_brazil.orders` AS o
JOIN
  `target_brazil.payments` AS p ON o.order_id = p.order_id
GROUP BY
  order_month,
  p.payment_type
ORDER BY
  order_month;
```

Query results

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

Row	order_month	payment_type	order_count
1	2016-09-01 00:00:00 UTC	credit_card	3
2	2016-10-01 00:00:00 UTC	credit_card	254
3	2016-10-01 00:00:00 UTC	voucher	23
4	2016-10-01 00:00:00 UTC	debit_card	2
5	2016-10-01 00:00:00 UTC	UPI	63
6	2016-12-01 00:00:00 UTC	credit_card	1
7	2017-01-01 00:00:00 UTC	voucher	61
8	2017-01-01 00:00:00 UTC	UPI	197
9	2017-01-01 00:00:00 UTC	credit_card	583
10	2017-01-01 00:00:00 UTC	debit_card	9

2. Count of orders based on the no. of payment installments

```
SELECT
  payment_installments,
  COUNT(*) AS order_count
FROM
  `target_brazil.payments`
GROUP BY
```

```
payment_installments;
```

JOB INFORMATION | RESULTS | JSON | EX

Row	payment_installment	order_count
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Actionable Insights (10 points)

1. Customer Segmentation:

- Actionable Insight: Target should create personalized marketing campaigns for different customer segments based on demographics and preferences to improve engagement. For example, they can send targeted offers to young, tech-savvy customers through digital channels, while offering exclusive discounts to loyal, high-spending customers.

2. Product Performance:

- Actionable Insight: Target should identify underperforming products and consider optimizing their assortment. For instance, they can analyze sales data to identify slow-moving items and make data-driven decisions to either reposition or discontinue them. Additionally, they can allocate more shelf space or marketing efforts to top-selling products.

3. Pricing Optimization:

- Actionable Insight: Target should conduct price elasticity analysis on various products to determine optimal pricing strategies. For instance, they can identify price-sensitive products and test different price points to find the sweet spot that maximizes sales and profitability. Additionally, they can introduce dynamic pricing for certain products based on demand and competitor pricing.

4. Promotions and Marketing Campaigns:

- Actionable Insight: Target should measure the effectiveness of different marketing channels and campaigns to allocate resources wisely. For example, if a particular social media platform generates a high return on investment, it can increase its advertising budget on that platform. They can also use customer segmentation data to target specific campaigns to relevant customer segments, increasing conversion rates.

5. Customer Journey and Experience:

- Actionable Insight: Target should invest in improving the online and in-store customer experience. For instance, they can analyze customer feedback and complaints to identify pain points in the customer journey and take appropriate actions to address them.

Implementing features like personalized recommendations, easy checkout processes, and efficient customer support can enhance customer satisfaction and loyalty.

6. Supply Chain Optimization:

- Actionable Insight: Target should analyze supply chain data to identify inefficiencies and optimize operations. For example, by tracking delivery times and transportation costs, they can optimize routes, negotiate better terms with suppliers, and improve inventory management to reduce stockouts and overstock situations.

7. Fraud Detection:

- Actionable Insight: Target should implement advanced fraud detection algorithms that analyze transaction data in real time. By monitoring transaction patterns and detecting anomalies, they can prevent fraudulent activities and minimize financial losses. They can also invest in secure payment gateways and employ machine learning models for continuous improvement.

8. Market Basket Analysis:

- Actionable Insight: Target should leverage market basket analysis to identify product associations and optimize product placement. For instance, by placing complementary products near each other, they can encourage customers to make additional purchases, ultimately increasing the average order value.

9. Customer Churn Prediction:

- Actionable Insight: Target should use machine learning algorithms to predict customer churn. By identifying customers at risk of leaving, they can implement targeted retention strategies such as personalized offers, loyalty programs, and proactive customer support to prevent churn and retain valuable customers.

10. Store Performance Analysis:

- Actionable Insight: Target should regularly analyze store-level metrics such as sales per square foot, foot traffic, and conversion rates. By identifying high-performing stores, they can share best practices across locations and allocate resources efficiently. They can also identify underperforming stores and take corrective actions such as remodeling, improved staffing, or local marketing campaigns.

Recommendations (10 points)

1. Customer Segmentation:

- Recommendation: Target should invest in data-driven customer segmentation to personalize marketing efforts. This includes leveraging customer data to create targeted campaigns, tailored offers, and personalized recommendations. By understanding the unique preferences and needs of different customer segments, Target can enhance customer engagement and loyalty.

2. Product Performance:

- Recommendation: Target should regularly analyze product performance metrics to optimize their product assortment. This involves identifying underperforming products and making informed decisions regarding their positioning, pricing, or potential discontinuation. Additionally, Target can capitalize on the success of top-selling products by allocating resources to promote and expand those product lines.

3. Pricing Optimization:

- Recommendation: Target should conduct a thorough price elasticity analysis to determine optimal pricing strategies. This involves identifying price-sensitive products and experimenting with different price points to find the balance between maximizing sales volume and maintaining profitability. Implementing dynamic pricing strategies and monitoring competitor pricing can also help Target stay competitive in the market.

4. Promotions and Marketing Campaigns:

- Recommendation: Target should track the effectiveness of marketing channels and campaigns to optimize their marketing spend. By leveraging customer segmentation data, Target can create targeted campaigns tailored to specific customer segments. They should also focus on channels that generate the highest return on investment and continually measure campaign performance to refine their marketing strategies.

5. Customer Journey and Experience:

- Recommendation: Target should prioritize improving the overall customer journey and experience across various touchpoints. This includes optimizing their online platforms, enhancing in-store experiences, and providing exceptional customer service. Implementing

personalized recommendations, seamless checkout processes, and responsive customer support can help enhance customer satisfaction and drive repeat business.

6. Supply Chain Optimization:

- Recommendation: Target should regularly analyze supply chain data to identify inefficiencies and optimize operations. This includes streamlining logistics, optimizing inventory management, and strengthening relationships with suppliers. By reducing delivery times, minimizing stockouts, and managing costs effectively, Target can improve overall operational efficiency and customer satisfaction.

7. Fraud Detection:

- Recommendation: Target should invest in robust fraud detection systems and security measures to protect customer transactions and minimize financial losses. By leveraging advanced algorithms and machine learning models, they can proactively identify and prevent fraudulent activities. Regular monitoring, continuous improvement of fraud detection mechanisms, and staying updated with emerging security technologies are essential.

8. Market Basket Analysis:

- Recommendation: Target should leverage market basket analysis insights to optimize product placement and improve cross-selling opportunities. By strategically placing complementary products together, they can encourage customers to make additional purchases and increase the average order value. This can be further enhanced by personalized product recommendations and targeted promotions based on the customer's purchase history.

9. Customer Churn Prediction:

- Recommendation: Target should utilize predictive analytics and machine learning models to identify customers at risk of churn. By proactively reaching out to those customers with personalized retention strategies, such as exclusive offers, loyalty rewards, and dedicated customer support, Target can increase customer retention and reduce churn.

10. Store Performance Analysis:

- Recommendation: Target should regularly evaluate store performance metrics to identify both high-performing and underperforming stores. This involves sharing best practices across locations, optimizing staffing levels, and tailoring marketing efforts to the specific needs of each store's local market. Additionally, Target can consider remodeling or renovating underperforming stores to enhance the overall shopping experience.

THANK YOU