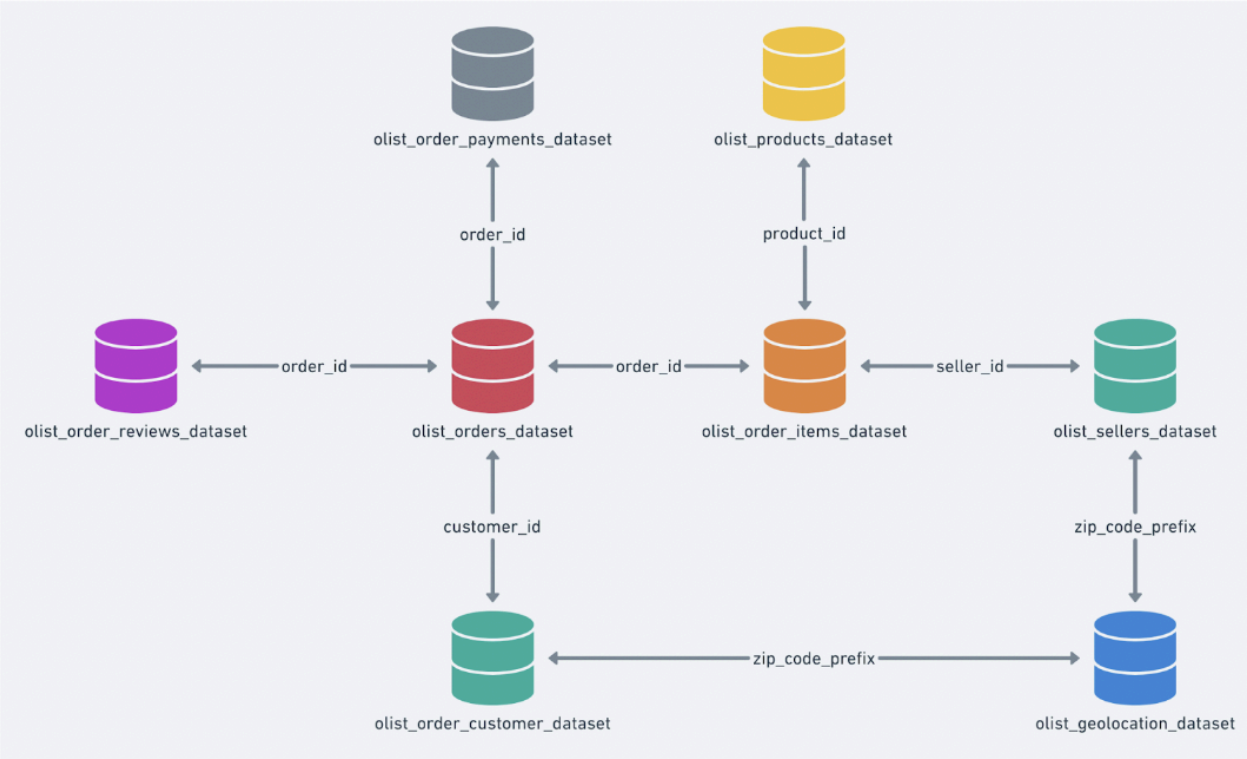
TARGET CASE STUDY

By Abha Sharma(abha.sharmaa111@gmail.com)

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.
This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

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



```
#Q.Import the dataset and do usual exploratory analysis steps like checking the
structure & characteristics of the dataset
```

1.Exploratory analysis

```
#Q.1.1Data type of all columns in the "customers" table
SELECT * FROM `Target.INFORMATION_SCHEMA.COLUMNS`;

SELECT
   column_name,
   data_type
FROM
  `Target.INFORMATION_SCHEMA.COLUMNS`
WHERE
   table_name = 'customers';
```

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

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

SELECT min(order_purchase_timestamp) as First_order,
      max(order_purchase_timestamp ) as last_order,
      date_diff(max(order_purchase_timestamp ),
      min(order_purchase_timestamp), day) as Difference_in_days
FROM `Target.orders`;
```

Row	First_order	Last_order	Difference_in_days
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

Time period ranges approximately 2 Years for the given data.

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

SELECT
 count(distinct customer_city) as City_count,
 count(distinct customer_state) as State_count FROM
`Target.customers` as c JOIN `Target.orders` as o
ON c.customer_id = o.customer_id;
```

Row	city_Count	state_count
1	4119	27

-- 2. In-depth Exploration:

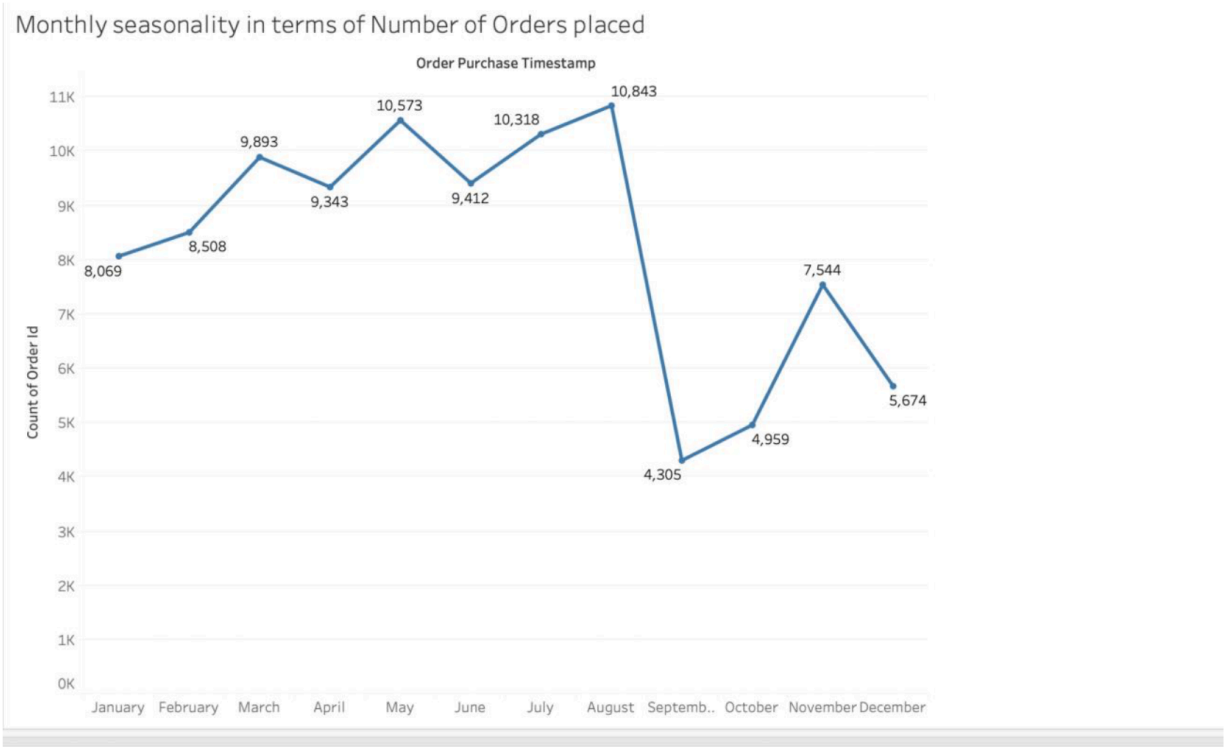
```
-- 1. Is there a growing trend in the no. of orders placed over the past years?
-- Growing trends considering year-on-year analysis for 2016, 2017, and 2018 isn't
very insightful here, assuming that the question is asking month-on-month analysis.

SELECT EXTRACT( Year from order_purchase_timestamp) AS order_year,
      EXTRACT( month from order_purchase_timestamp) AS order_month,
     COUNT(*) AS order_count
FROM `Target.orders`
GROUP BY
     order_year, order_month
ORDER BY
     order_year, order_month;
```

Row	order_year	order_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

-- From September 2016 to November 2017, there is a general upward trend in order counts, indicating overall growth in orders placed. However, this growth trend was disrupted in December 2017, and the orders count dropped, recovered back from January 2018 and was stable and dropped drastically again in September 2018.

```
-- 2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
SELECT
   EXTRACT( month from order_purchase_timestamp) AS order_month,
   COUNT(*) AS order_count
FROM `Target.orders`
```

```
GROUP BY
    order_month
ORDER BY
    Order_month;
```



-- There is a general trend of higher order counts in the middle months of the year (May, June, July, August), possibly due to summer holiday shopping, indicating that these months might be popular for shopping.

-- September has the lowest order count in this dataset. It might be worth investigating potential reasons for the sudden and significant drop in orders during this month.

-- 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

```sql
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'
       ELSE 'Night'
   END AS time_of_day,
   COUNT(*) AS order_count
FROM
   `Target.orders`
GROUP BY
   time_of_day
ORDER BY
   order_count DESC;
```

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

```sql
-- Most orders were placed during the afternoon followed by Night.
```

```sql
-- 3. . Evolution of E-commerce orders in the Brazil region:
-- 1. Get the month on month no. of orders placed in each state.
SELECT
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
   c.customer_state AS state,
   COUNT(*) AS order_count
FROM
   `Target.orders` AS o
```

```
JOIN
   `Target.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
   order_month, state
ORDER BY
    order_count DESC,order_month, state;
```

Row	order_month	state	order_count
1	8	SP	4982
2	5	SP	4632
3	7	SP	4381
4	6	SP	4104
5	3	SP	4047
6	4	SP	3967
7	2	SP	3357
8	1	SP	3351
9	11	SP	3012

```
-- 2. How are the customers distributed across all the states?
SELECT
   customer_state,
   count(*) as customer_count
FROM
   `Target.customers`
GROUP BY
   customer_state
ORDER BY
    customer_count desc;
```

Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140

Load more

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

```sql
-- 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.
WITH CTE AS (
   SELECT *
   FROM `Target.payments` AS p
   JOIN `Target.orders` AS o ON p.order_id = o.order_id
   WHERE EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
     AND EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND 2018
),
CTE2 AS (
   SELECT
       EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
       SUM(payment_value) AS cost
   FROM CTE
   GROUP BY year
   ORDER BY year
)
SELECT
   year,
   ROUND(cost, 2) AS cost,
   round(LEAD(cost) OVER (ORDER BY year),2) AS next_year_value,
   ROUND(((LEAD(cost) OVER (ORDER BY year) - cost) / cost) * 100, 2) AS percent_increase
FROM
   CTE2;
```

Row	year	cost	next_year_value	percent_increase
1	2017	3669022.12	8694733.84	136.98
2	2018	8694733.84	*null*	*null*

```
-- There is a 136.98 % increase in the cost of orders from 2017 to 2018.
```

```
-- 2. Calculate the Total & Average value of order price for each state.
SELECT
   c.customer_state,
   ROUND(SUM(oil.price), 2) AS total_price,
   ROUND(SUM(oil.price) / COUNT(DISTINCT oil.order_id), 2) AS average_price
FROM
   `Target.customers` AS c
JOIN
   `Target.orders` AS o ON c.customer_id = o.customer_id
JOIN
   `Target.order_items` AS oil ON oil.order_id = o.order_id
GROUP BY
   c.customer_state
ORDER BY
   C.customer_state;
```

Row	customer_state	total_price	average_price
1	AC	15982.95	197.32
2	AL	80314.81	195.41
3	AM	22356.84	152.09
4	AP	13474.3	198.15
5	BA	511349.99	152.28
6	CE	227254.71	171.25
7	DF	302603.94	142.4
8	ES	275037.31	135.82

Load more

```
-- 3.Calculate the Total & Average value of order freight for each state.
SELECT
   c.customer_state,
```

```
   ROUND(SUM(oil.price), 2) AS total_price,
   ROUND(SUM(oil.price) / COUNT(DISTINCT oil.order_id), 2) AS average_price
FROM
   `Target.customers` AS c
JOIN
   `Target.orders` AS o ON c.customer_id = o.customer_id
JOIN
   `Target.order_items` AS oil ON oil.order_id = o.order_id
GROUP BY
   c.customer_state
ORDER BY
   C.customer_state;
```

Row	customer_state	total_price	average_price
1	AC	15982.95	197.32
2	AL	80314.81	195.41
3	AM	22356.84	152.09
4	AP	13474.3	198.15
5	BA	511349.99	152.28
6	CE	227254.71	171.25
7	DF	302603.94	142.4
8	ES	275037.31	135.82
9	GO	294591.95	146.78

```
-- 5. Analysis based on sales, freight and delivery time.
-- 1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
-- Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
-- Do this in a single query.
-- 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
SELECT
   order_id,
   order_purchase_timestamp,
```

```sql
   order_delivered_customer_date,
   order_estimated_delivery_date,
   TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
   TIMESTAMP_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY)
AS diff_estimated_delivery
FROM
   `Target.orders`
WHERE
   order_status = 'delivered';

-- 87182 orders out of 96478 were delivered before the estimated delivery date.

-- 2. Find out the top 5 states with the highest & lowest average freight value.
WITH CTE AS (
   SELECT
       c.customer_state,
       ROUND(SUM(oil.freight_value) / COUNT(DISTINCT oil.order_id), 2) AS
average_freight_value
   FROM
       `Target.customers` AS c
   JOIN
       `Target.orders` AS o ON c.customer_id = o.customer_id
   JOIN
       `Target.order_items` AS oil ON oil.order_id = o.order_id
   GROUP BY
       c.customer_state
)
-- Top 5 Lowest Average Freight
(SELECT
   'Top 5 Lowest Average Freight' AS category,
   customer_state,
   average_freight_value
FROM
   CTE
ORDER BY
```

```
    average_freight_value
LIMIT 5)

UNION ALL

-- Top 5 Highest Average Freight
(SELECT
    'Top 5 Highest Average Freight' AS category,
    customer_state,
    average_freight_value
FROM
    CTE
ORDER BY
    average_freight_value DESC
LIMIT 5);
```

Row	category	customer_state	average_freight_valu
1	Top 5 Highest Average Freight	RR	48.59
2	Top 5 Highest Average Freight	PB	48.35
3	Top 5 Highest Average Freight	RO	46.22
4	Top 5 Highest Average Freight	AC	45.52
5	Top 5 Highest Average Freight	PI	43.04
6	Top 5 Lowest Average Freight	SP	17.37
7	Top 5 Lowest Average Freight	MG	23.46
8	Top 5 Lowest Average Freight	PR	23.58
9	Top 5 Lowest Average Freight	DF	23.82
10	Top 5 Lowest Average Freight	RJ	23.95

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

WITH CTE AS (
    SELECT
        c.customer_state,
        ROUND(SUM(TIMESTAMP_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)) / COUNT(DISTINCT c.customer_id), 2) AS
average_delivery_time
```

```sql
    FROM
        `Target.customers` AS c
    JOIN
        `Target.orders` AS o ON c.customer_id = o.customer_id
    GROUP BY
        c.customer_state
)
-- Top 5 Lowest Average Delivery Time
(SELECT
    'Top 5 Lowest Average Delivery Time' AS category,
    customer_state,
    average_delivery_time
FROM
    CTE
ORDER BY
    average_delivery_time
LIMIT 5)

UNION ALL

-- Top 5 Highest Average Delivery Time
(SELECT
    'Top 5 Highest Average Delivery Time' AS category,
    customer_state,
    average_delivery_time
FROM
    CTE
ORDER BY
    average_delivery_time DESC
LIMIT 5);
```

Row	category	customer_state	average_delivery_tim
1	Top 5 Highest Average Delivery...	AP	26.34
2	Top 5 Highest Average Delivery...	RR	25.83
3	Top 5 Highest Average Delivery...	AM	25.46
4	Top 5 Highest Average Delivery...	AL	23.11
5	Top 5 Highest Average Delivery...	PA	22.62
6	Top 5 Lowest Average Delivery ...	SP	8.05
7	Top 5 Lowest Average Delivery ...	PR	11.25
8	Top 5 Lowest Average Delivery ...	MG	11.27
9	Top 5 Lowest Average Delivery ...	DF	12.16
10	Top 5 Lowest Average Delivery ...	SC	14.12

```
-- 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.
SELECT
   c.customer_state,
   ROUND(SUM(TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY))
         / COUNT(DISTINCT c.customer_id), 2) AS diff_estimated_delivery
FROM
   `Target.customers` AS c
JOIN
   `Target.orders` AS o ON c.customer_id = o.customer_id
GROUP BY
   c.customer_state
ORDER BY
   diff_estimated_delivery DESC
LIMIT 5;
```

Row	customer_state	diff_estimated_delive
1	AC	19.52
2	AP	18.46
3	RO	18.38
4	AM	18.23
5	RR	14.63

```sql
-- 6. Analysis based on the payments:
-- 1. Find the month on month no. of orders placed using different payment types.
SELECT
   p.payment_type,
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS Month,
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS Year,
   COUNT(*) AS order_count
FROM
   `Target.payments` AS p
JOIN
   `Target.orders` AS o ON p.order_id = o.order_id
GROUP BY
   p.payment_type, Year, Month
ORDER BY
   Year, Month, p.payment_type;
```

Row	payment_type	Month	Year	order_count
1	credit_card	9	2016	3
2	UPI	10	2016	63
3	credit_card	10	2016	254
4	debit_card	10	2016	2
5	voucher	10	2016	23
6	credit_card	12	2016	1
7	UPI	1	2017	197
8	credit_card	1	2017	583

Load more

```sql
-- Most used payment method was credit_card followed by UPI

-- 2. Find the no. of orders placed on the basis of the payment installments that have been paid.
SELECT
   payment_installments,
   COUNT(*) AS order_count
FROM
   `Target.payments`
WHERE
   payment_installments > 1
```

```
GROUP BY
   Payment_installments;
```

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