

Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

What does 'good' look like?

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.

<input type="checkbox"/>	Field name	Type
<input type="checkbox"/>	customer_id	STRING
<input type="checkbox"/>	customer_unique_id	STRING
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER
<input type="checkbox"/>	customer_city	STRING
<input type="checkbox"/>	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
time_range
FROM `target.orders`;
```

Row	first_order	last_order	time_range
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

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

```
SELECT
    COUNT(DISTINCT customer_city) AS no_of_cities,
    COUNT(DISTINCT customer_state) AS no_of_states
FROM `target.customers`;
```

Row	no_of_cities	no_of_states
1	4119	27

2. In-depth Exploration:

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

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

Row	year	total_orders
1	2016	329
2	2017	45101
3	2018	54011

Based on the query results, we can conclude that there is a **growing trend** in the number of orders placed over the past years.

1. There is a **substantial increase** in the number of orders from 2016 to 2017, indicating rapid growth.
 2. The trend continued with another increase from 2017 to 2018, though at a slower rate compared to the previous jump.
 3. Overall, the data shows a **positive growth trend** in the number of orders placed each year, which is a promising indicator of business expansion and increased customer engagement.
-
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(order_id) AS total_orders
FROM `target.orders`
GROUP BY order_month
ORDER BY order_month;
```

Row	order_month	total_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

Monthly Seasonality in Orders:

Yes, there is a clear monthly seasonality in the number of orders placed. The peak months are August (10,843) and May (10,573), while the lowest orders are observed in September (4,305) and October (4,959). There is a noticeable surge in orders from March to August, followed by a decline towards the end of the year.

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

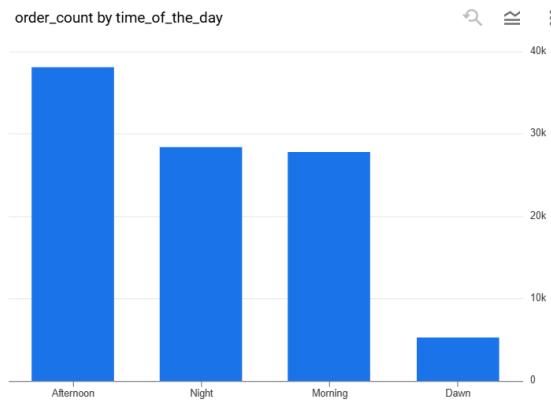
```
WITH time_period AS (
    SELECT
        order_id,
        CASE
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN
                'Morning'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
                'Afternoon'
            ELSE 'Night'
        END AS time_of_the_day
    FROM `target.orders`)
SELECT
    time_of_the_day,
    COUNT(order_id) AS order_count
FROM time_period
GROUP BY time_of_the_day
ORDER BY order_count DESC;
```

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

Recommendation to Business:

1. Ensure product availability in the afternoon
2. Provide Flash discounts at peak hours
3. Reduce operational costs
4. Personalized notifications about offers

5. Ensure no server down in peak hours



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

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

```
WITH order_data AS (
  SELECT
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
    c.customer_state,
    COUNT(o.order_id) AS no_of_orders
  FROM `target.orders` o
  LEFT JOIN `target.customers` c
    ON o.customer_id = c.customer_id
  GROUP BY order_month, c.customer_state
),
month_on_month AS (
  SELECT
    order_month,
    customer_state,
    no_of_orders,
    no_of_orders - LAG(no_of_orders, 1) OVER (
      PARTITION BY customer_state ORDER BY order_month
    ) AS month_on_month_difference
  FROM order_data
)
SELECT *
FROM month_on_month
ORDER BY order_month, customer_state;
```

Row	order_month	customer_state	no_of_orders	month_on_month_difference
1	1	AC	8	nuli
2	1	AL	39	nuli
3	1	AM	12	nuli
4	1	AP	11	nuli
5	1	BA	264	nuli
6	1	CE	99	nuli
7	1	DF	151	nuli
8	1	ES	159	nuli
9	1	GO	164	nuli
10	1	MA	66	nuli
Row	order_month	customer_state	no_of_orders	month_on_month_difference
310	12	PA	58	-12
311	12	PB	37	7
312	12	PE	103	-23
313	12	PI	23	-8
314	12	PR	271	-107
315	12	RJ	783	-265
316	12	RN	30	-14
317	12	RO	11	-6
318	12	RS	283	-139
319	12	SC	193	-110
320	12	SE	20	-7
321	12	SP	2357	-655
322	12	TO	14	-3

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

```

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

```

Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

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

```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    SUM(i.price) AS total_price,
    SUM(i.freight_value) AS total_freight_value
FROM `target.orders` o
JOIN `target.orders_items` i ON o.order_id = i.order_id
GROUP BY year
ORDER BY year;
```

Row	year	total_price	total_freight_value
1	2016	49785.92000000...	7397.28999999...
2	2017	6155806.980000...	986865.450000...
3	2018	7386050.800000...	1257646.799999...

- Yearly Money Movement: The total revenue generated from orders has shown a significant increase from 2016 to 2018, indicating a rapid growth in e-commerce transactions over the years.
- Freight Impact: The freight value also increased, suggesting either higher shipping volumes or more expensive shipping rates over time.

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 t AS (
    SELECT i.price,
        i.freight_value,
        EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
        EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month
    FROM `target.orders` o
    JOIN `target.orders_items` i ON o.order_id = i.order_id
    WHERE EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
)
,
yearly_totals AS (
    SELECT order_year,
        SUM(price) AS total_price,
        SUM(freight_value) AS total_freight_value
    FROM t
    GROUP BY order_year
)
SELECT y2017.total_price AS total_price_2017,
    y2018.total_price AS total_price_2018,
    ROUND(((y2018.total_price - y2017.total_price) / y2017.total_price) * 100, ) AS
percentage_increase
FROM (
    SELECT total_price
    FROM yearly_totals
    WHERE order_year = 2017
```

```
) AS y2017
CROSS JOIN (
    SELECT total_price
    FROM yearly_totals
    WHERE order_year = 2018
) AS y2018;
```

Row	total_payment_2017	total_payment_2018	percentage_increase
1	3669022.120000...	8694733.840000...	136.98

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

```
SELECT c.customer_state,
    ROUND(AVG(COALESCE(i.price, 0)), 2) AS avg_order_price,
    ROUND(SUM(COALESCE(i.price, 0)), 2) AS total_order_price
FROM `target.customers` c
LEFT JOIN `target.orders` o
ON c.customer_id = o.customer_id
LEFT JOIN `target.orders_items` i
ON o.order_id = i.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state ASC;
```

Row	customer_state ▾	avg_order_price ▾	total_order_price ▾
1	AC	173.73	15982.95
2	AL	180.08	80314.81
3	AM	134.68	22356.84
4	AP	164.32	13474.3
5	BA	133.83	511349.99
6	CE	152.83	227254.71
7	DF	124.99	302603.94
8	ES	121.48	275037.31
9	GO	125.57	294591.95
10	MA	143.98	119648.22

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

```
SELECT c.customer_state,
    ROUND(AVG(COALESCE(i.freight_value, 0)), 2) AS avg_freight_value,
    ROUND(SUM(COALESCE(i.freight_value, 0)), 2) AS total_freight_value
FROM `target.customers` c
LEFT JOIN `target.orders` o
ON c.customer_id = o.customer_id
LEFT JOIN `target.orders_items` i
```

```

ON o.order_id = i.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state ASC;

```

Row	customer_state	avg_freight_value	total_freight_value
1	AC	40.07	3686.75
2	AL	35.68	15914.59
3	AM	33.01	5478.89
4	AP	34.01	2788.5
5	BA	26.21	100156.68
6	CE	32.52	48351.59
7	DF	20.91	50625.5
8	ES	21.98	49764.6
9	GO	22.64	53114.98
10	MA	37.93	31523.77

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.
Do this in a single query.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- time_to_deliver** = `order_delivered_customer_date - order_purchase_timestamp`
- diff_estimated_delivery** = `order_delivered_customer_date - order_estimated_delivery_date`

```

SELECT order_id,
       DATE(order_purchase_timestamp) AS order_purchase_date,
       DATE(order_delivered_customer_date) AS order_delivered_date,
       DATE(order_estimated_delivery_date) AS order_estimated_date,
       DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp), DAY)
AS time_to_deliver,
       DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_estimated_delivery_date),
DAY) AS diff_estimated_delivery
FROM `target.orders`
ORDER BY order_id ASC;

```

Row	order_id	order_purchase_date	order_delivered_date	order_estimated_date	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d...	2017-09-13	2017-09-20	2017-09-29	7	-9
2	00018f77f2f0320c...	2017-04-26	2017-05-12	2017-05-15	16	-3
3	000229ec398224ef...	2018-01-14	2018-01-22	2018-02-05	8	-14
4	00024acbcdf0a6da...	2018-08-08	2018-08-14	2018-08-20	6	-6
5	00042b26cf59d7ce...	2017-02-04	2017-03-01	2017-03-17	25	-16
6	00048cc3ae777c6...	2017-05-15	2017-05-22	2017-06-06	7	-15
7	00054e8431b9d76...	2017-12-10	2017-12-18	2018-01-04	8	-17
8	000576fe3931984...	2018-07-04	2018-07-09	2018-07-25	5	-16
9	0005a1a1728c9d7...	2018-03-19	2018-03-29	2018-03-29	10	0
10	0005f50442cb953...	2018-07-02	2018-07-04	2018-07-23	2	-19

- ✓ **time_to_deliver:** Calculates the number of days taken to deliver each order from the purchase date.
- ✓ **diff_estimated_delivery:** Calculates the difference (in days) between the actual and estimated delivery dates.
- A positive value means **delayed delivery**.
- A negative value means **early delivery**.
- A zero value means **on-time delivery**.

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

```
WITH avg_freight AS (
  SELECT c.customer_state,
         ROUND(AVG(i.freight_value), 2) AS avg_freight_value
    FROM `target.customers` c
   LEFT JOIN `target.orders` o ON c.customer_id = o.customer_id
   JOIN `target.orders_items` i ON o.order_id = i.order_id
   GROUP BY c.customer_state
)
```

```
-- Top 5 states with the highest average freight value
SELECT customer_state, avg_freight_value
FROM avg_freight
ORDER BY avg_freight_value DESC
LIMIT 5;
```

Row	customer_state	avg_freight_value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

```
-- Top 5 states with the lowest average freight value
SELECT customer_state, avg_freight_value
FROM avg_freight
ORDER BY avg_freight_value ASC
LIMIT 5;
```

Row	customer_state	avg_freight_value
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

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

```
-- Top 5 states with the highest average delivery time
```

```
WITH delivery_details AS (
    SELECT c.customer_state,
        DATE_DIFF(DATE(o.order_delivered_customer_date),
DATE(o.order_purchase_timestamp), DAY) AS delivery_time
    FROM `target.customers` c
    JOIN `target.orders` o ON c.customer_id = o.customer_id
)
```

```
SELECT customer_state,
    ROUND(AVG(delivery_time), 2) AS avg_delivery_time
FROM delivery_details
GROUP BY customer_state
ORDER BY avg_delivery_time DESC
LIMIT 5;
```

Row	customer_state	avg_delivery_time
1	RR	29.34
2	AP	27.18
3	AM	26.36
4	AL	24.5
5	PA	23.73

```
-- Top 5 states with the lowest average delivery time
```

```
WITH delivery_details AS (
    SELECT c.customer_state,
        DATE_DIFF(DATE(o.order_delivered_customer_date),
DATE(o.order_purchase_timestamp), DAY) AS delivery_time
    FROM `target.customers` c
    JOIN `target.orders` o ON c.customer_id = o.customer_id
)
```

```
SELECT customer_state,
    ROUND(AVG(delivery_time), 2) AS avg_delivery_time
```

```

FROM delivery_details
GROUP BY customer_state
ORDER BY avg_delivery_time ASC
LIMIT 5;

```

Row	customer_state	avg_delivery_time
1	SP	8.7
2	PR	11.94
3	MG	11.95
4	DF	12.9
5	SC	14.91

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(AVG(DATE_DIFF(DATE(o.order_estimated_delivery_date),
DATE(o.order_delivered_customer_date), DAY)), 2) AS avg_diff_delivery_time
FROM `target.customers` c
JOIN `target.orders` o ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_diff_delivery_time DESC
LIMIT 5;

```

Row	customer_state	avg_diff_delivery_time
1	AC	20.72
2	RO	20.1
3	AP	19.69
4	AM	19.57
5	RR	17.29

6. Analysis based on the payments:

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

```

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

```

Row	year	month	payment_type	no_of_orders
1	2016	9	credit_card	3
2	2016	10	voucher	23
3	2016	10	credit_card	254
4	2016	10	UPI	63
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	583
8	2017	1	UPI	197
9	2017	1	debit_card	9
10	2017	1	voucher	61

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

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

Row	installments_paid	no_of_orders
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

7. List down any valuable insights that you find during the analysis and provide some action items from the company's perspective in order to improve the current situation.

Insights:

- Order Trend: The number of orders has shown a significant increase from 2016 to 2018, indicating growing popularity and adoption of e-commerce.
 - ✓ Action: Continue investing in marketing to maintain this growth trajectory.

2. Monthly Seasonality: There is a noticeable peak in order volume during the months of May to August. The lowest order volumes occur in September and October.
 - ✓ Action: Plan marketing campaigns and discounts during low-demand months to boost sales.
3. Time of Day Analysis: Most orders are placed during the Afternoon (13-18 hrs) and Night (19-23 hrs), indicating higher customer activity during these periods.
 - ✓ Action: Enhance server capacity and customer support during peak hours to ensure smooth order processing.
4. Freight Charges Variation: States with higher average freight charges could indicate logistical challenges or longer distances.
 - ✓ Action: Reassess logistics strategies for high-cost states to optimize freight expenses.
5. Payment Preferences: Customers overwhelmingly prefer single installment payments, but a significant portion also opts for 2 or 3 installments.
 - ✓ Action: Offer installment-based promotions or discounts to encourage more purchases, especially for higher-value items.
6. Delivery Timeliness: There are occasional delays where actual delivery time exceeds estimated dates.
 - ✓ Action: Work with logistics partners to improve accuracy in estimated delivery dates and reduce delays.