# BUSINESS CASE : TARGET SQL

**Srinivas Bhairi**

**Batch : DSML'24**

## 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.

```
SELECT column_name,data_type
FROM  `scaler-dsml-sql-444310.Target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
```

| column_name | data_type |
|---|---|
| customer_id | STRING |
| customer_unique_id | STRING |
| customer_zip_code_prefix | INT64 |
| customer_city | STRING |
| customer_state | STRING |

**In-sights :**

**Data Understanding: Helps in understanding the structure of the `customers` table.**

**Schema Validation: Useful when writing queries to ensure correct data types are used.**

**Debugging: If queries fail due to type mismatches, this helps check the expected data type.**

**Data Transformation: Knowing data types helps in performing correct aggregations and conversions.**

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

```
select min(order_purchase_timestamp) as lowest_purchase_timestamp,
       max(order_purchase_timestamp) as highest_purchase_timestamp
from `Target.orders`
```

| lowest_purchase_timestamp | highest_purchase_timestamp |
|---|---|
| 2016-09-04 21:15:19.000000 UTC | 2018-10-17 17:30:18.000000 UTC |

**In-sight :**

**Understanding Order Timeline: Helps determine the date range of available order data.**

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

```
select count(distinct c.customer_city) as no_of_cities,
        count(distinct c.customer_state) as no_of_states
from `Target.customers` c
inner join `Target.orders` o ON c.customer_id = o.customer_id
```

| no_of_cities | no_of_states |
|---|---|
| 4119 | 27 |

**In-sights:**

- Customer Reach: The number of cities and states where orders were placed helps understand the geographic spread of customers.
- Market Penetration: If the number of unique states is low, there might be potential markets to expand into.
- Regional Popularity: If a large number of cities contribute to the orders, the business has a diverse customer base.

**—---------------------------------------------------------------------------------**

# 2.In-depth Exploration:

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

```
WITH yearly_orders AS (
    SELECT
        EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
        COUNT(order_id) AS total_orders
    FROM `Target.orders`
    GROUP BY order_year
)

SELECT
    order_year,
    total_orders,
    LAG(total_orders) OVER (ORDER BY order_year) AS prev_year_orders,
    ROUND(((total_orders - LAG(total_orders) OVER (ORDER BY order_year)) /
LAG(total_orders) OVER (ORDER BY order_year)) * 100, 2) AS growth_percentage
FROM yearly_orders
order by order_year
```

| order_year | total_orders | prev_year_orders | growth_percentage |
|---|---|---|---|
| 2016 | 329 | null | null |
| 2017 | 45101 | 329 | 13,608.51 |
| 2018 | 54011 | 45101 | 19.76 |

**Identifying Growth Trends**

- **A positive growth percentage indicates increasing order volume year-over-year, which suggests strong customer demand & business expansion.**
- **A negative growth percentage signals a decline, potentially due to market competition, pricing strategies, or operational issues.**

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

```
SELECT
    FORMAT_DATE("%B",order_purchase_timestamp) AS order_month,
    COUNT(order_id) AS total_orders
FROM `Target.orders`
GROUP BY order_month
order by total_orders
```

| order_month | total_orders |
|---|---|
| September | 4305 |
| October | 4959 |
| December | 5674 |
| November | 7544 |
| January | 8069 |
| February | 8508 |
| April | 9343 |
| June | 9412 |
| March | 9893 |
| July | 10318 |
| May | 10573 |
| August | 10843 |

**Seasonality Trends:**

- If certain months (e.g., May,August) have significantly higher orders, it might indicate seasonal demand, possibly due to holidays, Independence day sales.
- Conversely, months with lower order counts might indicate off-peak periods where fewer customers are purchasing.

## 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
WITH final AS (
    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 day_time,
        COUNT(order_id) AS no_of_orders
    FROM `Target.orders`
    GROUP BY day_time
)

SELECT day_time,
       no_of_orders
FROM final
ORDER BY no_of_orders DESC
LIMIT 1;
```

| day_time | most_orders |
|---|---|
| Afternoon | 38135 |

**Peak Order Time**

- The result will show which time period has the highest order volume.
- If Afternoon or Night has the most orders, it suggests customers are more active during these periods.
- If Morning leads, it may indicate business-related orders, possibly from B2B clients.

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

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

```
select c.customer_state,
       extract(month from o.order_purchase_timestamp) as order_month,
       count(o.order_id) as no_of_orders

from `Target.orders` o
inner join `Target.customers` c
ON o.customer_id = c.customer_id
group by c.customer_state , order_month
order by c.customer_state , order_month
```

| customer_state | order_month | no_of_orders |
|---|---:|---:|
| AC | 1 | 8 |
| AC | 2 | 6 |
| AC | 3 | 4 |
| AC | 4 | 9 |
| AC | 5 | 10 |
| AC | 6 | 7 |
| AC | 7 | 9 |
| AC | 8 | 7 |
| AC | 9 | 5 |
| AC | 10 | 6 |
| AC | 11 | 5 |
| AC | 12 | 5 |
| AL | 1 | 39 |
| AL | 2 | 39 |
| AL | 3 | 40 |
| AL | 4 | 51 |
| AL | 5 | 46 |
| AL | 6 | 34 |

|  |  |  |
|---|---|---|
| AL | 7 | 40 |
| AL | 8 | 34 |
| AL | 9 | 20 |
| AL | 10 | 30 |
| AL | 11 | 26 |
| AL | 12 | 14 |

**Seasonal Demand Patterns :**

- **Some states may see higher order volumes in specific months (e.g., holiday seasons or festivals).**
- **If orders peak in November/December, it could be due to holiday shopping trends.**
- **If orders drop in certain months, the company can plan promotions to boost sales.**

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

```
select customer_state,
       count(distinct customer_id) as no_of_customers
from `Target.customers`
group by customer_state
order by customer_state
```

| customer_state | no_of_customers |
|---|---|
| AC | 81 |
| AL | 413 |
| AM | 148 |
| AP | 68 |
| BA | 3380 |
| CE | 1336 |
| DF | 2140 |
| ES | 2033 |
| GO | 2020 |
| MA | 747 |
| MG | 11635 |
| MS | 715 |
| MT | 907 |
| PA | 975 |
| PB | 536 |
| PE | 1652 |
| PI | 495 |

| | |
|---|---:|
| PR | 5045 |
| RJ | 12852 |
| RN | 485 |
| RO | 253 |
| RR | 46 |
| RS | 5466 |
| SC | 3637 |
| SE | 350 |
| SP | 41746 |
| TO | 280 |

**Identifying Key Market Regions**

- **States with the highest number of customers are high-priority markets for sales and marketing efforts.**
- **These states might need more warehouse locations to optimize shipping efficiency.**
- **States with a low number of customers indicate untapped markets.**

## 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).

```
with final as
(select sum(p.payment_value) as total_value,
       EXTRACT(year from o.order_purchase_timestamp) as order_year

from `Target.payments` p
inner join `Target.orders` o
ON o.order_id = p.order_id
where extract(month from o.order_purchase_timestamp) between 1 and 8
and EXTRACT(year from o.order_purchase_timestamp) in (2017,2018)
group by order_year
order by order_year)


select order_year,
       total_value,
       lag(total_value)over(order by order_year) as prev_value,
       ROUND(((total_value - LAG(total_value) OVER (ORDER BY order_year))/
LAG(total_value) OVER (ORDER BY order_year)) * 100, 2) as percentage_growth
from final
order by order_year
```

| order_year | total_value | prev_value | percentage_growth |
|---|---|---|---|
| 2017 | 3,669,022.12 | null | null |
| 2018 | 8,694,733.84 | 3,669,022.12 | 136.98 |

**If 2018 has higher revenue, it means the company scaled up successfully.**
**If revenue dropped in 2018, potential reasons could include:**

- **Market saturation**
- **Increased competition**
- **Operational inefficiencies**
- **Customer retention issues**
-

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

```
with final as
(select c.customer_state,
       count(distinct o.order_id) as no_of_orders,
       round(sum(i.price),2) as Total_value

from `Target.customers` c
inner join `Target.orders` o
ON c.customer_id = o.customer_id
inner join `Target.order_items` i
ON o.order_id = i.order_id
group by c.customer_state
)

select *,
       round((final.Total_value / final.no_of_orders),2) as avg_order_price
from final
order by customer_state
```

| customer_state | no_of_orders | Total_value | avg_order_price |
|---|---:|---:|---:|
| AC | 81 | 15,982.95 | 197.32 |
| AL | 411 | 80,314.81 | 195.41 |
| AM | 147 | 22,356.84 | 152.09 |
| AP | 68 | 13,474.30 | 198.15 |
| BA | 3358 | 511,349.99 | 152.28 |
| CE | 1327 | 227,254.71 | 171.25 |
| DF | 2125 | 302,603.94 | 142.4 |
| ES | 2025 | 275,037.31 | 135.82 |
| GO | 2007 | 294,591.95 | 146.78 |
| MA | 740 | 119,648.22 | 161.69 |
| MG | 11544 | 1,585,308.03 | 137.33 |
| MS | 709 | 116,812.64 | 164.76 |
| MT | 903 | 156,453.53 | 173.26 |
| PA | 970 | 178,947.81 | 184.48 |
| PB | 532 | 115,268.08 | 216.67 |
| PE | 1648 | 262,788.03 | 159.46 |
| PI | 493 | 86,914.08 | 176.3 |
| PR | 4998 | 683,083.76 | 136.67 |
| RJ | 12762 | 1,824,092.67 | 142.93 |
| RN | 482 | 83,034.98 | 172.27 |
| RO | 247 | 46,140.64 | 186.8 |
| RR | 46 | 7,829.43 | 170.21 |
| RS | 5432 | 750,304.02 | 138.13 |
| SC | 3612 | 520,553.34 | 144.12 |
| SE | 345 | 58,920.85 | 170.79 |
| SP | 41375 | 5,202,955.05 | 125.75 |
| TO | 279 | 49,621.74 | 177.86 |

**In-sights :**

**1. Identify High-Value States for Focused Marketing**
**2. Improve Performance in Low-Revenue States**

**Improvement:**

- **If the order_items table has product categories, analyzing which categories drive high avg order value can be helpful.**

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

```
with final as
(select c.customer_state as state,
       count(distinct o.order_id) as no_of_orders,
       round(sum(i.freight_value),2) as Total_freight_value

from `Target.customers` c
inner join `Target.orders` o
ON c.customer_id = o.customer_id
inner join `Target.order_items` i
ON o.order_id = i.order_id
group by c.customer_state
)


select *,
       round((final.Total_freight_value / final.no_of_orders),2) as avg_order_price
from final
order by state
```

| state | no_of_orders | Total_freight_value | avg_order_price |
|---|---:|---:|---:|
| AC | 81 | 3,686.75 | 45.52 |
| AL | 411 | 15,914.59 | 38.72 |
| AM | 147 | 5,478.89 | 37.27 |
| AP | 68 | 2,788.50 | 41.01 |
| BA | 3358 | 100,156.68 | 29.83 |
| CE | 1327 | 48,351.59 | 36.44 |
| DF | 2125 | 50,625.50 | 23.82 |
| ES | 2025 | 49,764.60 | 24.58 |
| GO | 2007 | 53,114.98 | 26.46 |
| MA | 740 | 31,523.77 | 42.6 |
| MG | 11544 | 270,853.46 | 23.46 |
| MS | 709 | 19,144.03 | 27 |
| MT | 903 | 29,715.43 | 32.91 |
| PA | 970 | 38,699.30 | 39.9 |
| PB | 532 | 25,719.73 | 48.35 |
| PE | 1648 | 59,449.66 | 36.07 |
| PI | 493 | 21,218.20 | 43.04 |
| PR | 4998 | 117,851.68 | 23.58 |
| RJ | 12762 | 305,589.31 | 23.95 |

| | | | |
|---|---:|---:|---:|
| RN | 482 | 18,860.10 | 39.13 |
| RO | 247 | 11,417.38 | 46.22 |
| RR | 46 | 2,235.19 | 48.59 |
| RS | 5432 | 135,522.74 | 24.95 |
| SC | 3612 | 89,660.26 | 24.82 |
| SE | 345 | 14,111.47 | 40.9 |
| SP | 41375 | 718,723.07 | 17.37 |
| TO | 279 | 11,732.68 | 42.05 |

**Freight Costs Vary Across States :**

- **States with higher total freight costs indicate:**
  - **A larger number of orders**
  - **Higher shipping costs due to distance, logistics challenges, or fewer fulfillment centers**

-----------------------------------------------------------------------------------

## 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_delivered_customer_date - order_estimated_delivery_date**

```
select order_id,
       date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
delivery_time,
       date_diff(order_delivered_customer_date,order_estimated_delivery_date,day)
as diff_estimated_delivery
from `Target.orders`
```

| order_id | delivery_time | diff_estimated_delivery |
|---|---:|---:|
| 1950d777989f6a877539f53795b4c3c3 | 30 | 12 |
| 2c45c33d2f9cb8ff8b1c86cc28c11c30 | 30 | -28 |
| 65d1e226dfaeb8cdc42f665422522d14 | 35 | -16 |
| 635c894d068ac37e6e03dc54eccb6189 | 30 | -1 |
| 3b97562c3aee8bdedcb5c2e45a50d5e1 | 32 | 0 |
| 68f47f50f04c4cb6774570cfde3a9aa7 | 29 | -1 |
| 276e9ec344d3bf029ff83a161c6b3ce9 | 43 | 4 |
| 54e1a3c2b97fb0809da548a59f64c813 | 40 | 4 |
| fd04fa4105ee8045f6a0139ca5b49f27 | 37 | 1 |
| 302bb8109d097a9fc6e9cefc5917d1f3 | 33 | 5 |
| 66057d37308e787052a32828cd007e58 | 38 | 6 |

**1.Actual Delivery Time**

- **date_diff(order_delivered_customer_date, order_purchase_timestamp, day)**
- **Measures the total days taken for delivery after order placement.**

**2. Delivery Speed vs. Estimated Date**

- **date_diff(order_delivered_customer_date, order_estimated_delivery_date, day)**
- **Positive values → Delivered late (past the estimated date)**
- **Negative values → Delivered early (faster than expected)**
- **Zero value → Delivered on the estimated date**

**3.Improve Delivery Time Predictions**

**4.Recognize Fast Deliveries**

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

```
WITH final AS (
    SELECT
        c.customer_state AS state,
        COUNT(DISTINCT o.order_id) AS no_of_orders,
        SUM(i.freight_value) AS total_freight_value,
        ROUND(SUM(i.freight_value) / COUNT(DISTINCT o.order_id), 2) AS
avg_freight_value
    FROM `Target.customers` c
    INNER JOIN `Target.orders` o ON c.customer_id = o.customer_id
    INNER JOIN `Target.order_items` i ON i.order_id = o.order_id
    GROUP BY c.customer_state
),

highest AS (
    SELECT state, avg_freight_value,
           ROW_NUMBER() OVER (ORDER BY avg_freight_value DESC) AS ranking
    FROM final
    LIMIT 5
),



lowest AS (
    SELECT state, avg_freight_value,
           ROW_NUMBER() OVER (ORDER BY avg_freight_value ASC) AS ranking
    FROM final
    LIMIT 5
)

SELECT state, avg_freight_value, ranking, 'Highest' AS type
FROM highest
UNION ALL
SELECT state, avg_freight_value, ranking, 'Lowest' AS type
FROM lowest
ORDER BY type , ranking
```

| state | avg_freight_value | ranking | type |
|---|---:|---:|---|
| RR | 48.59 | 1 | Highest |
| PB | 48.35 | 2 | Highest |
| RO | 46.22 | 3 | Highest |
| AC | 45.52 | 4 | Highest |
| PI | 43.04 | 5 | Highest |
| SP | 17.37 | 1 | Lowest |
| MG | 23.46 | 2 | Lowest |
| PR | 23.58 | 3 | Lowest |
| DF | 23.82 | 4 | Lowest |
| RJ | 23.95 | 5 | Lowest |

### 1. Highest Freight Cost States

- These states have high shipping costs per order.
- Possible reasons:
  - Distant locations from warehouses.
  - Higher demand for express delivery.
  - Limited logistics options leading to higher costs.

### 2. Lowest Freight Cost States

- These states have low shipping charges per order.
- Possible reasons:
  - Closer proximity to sellers and warehouses.
  - Efficient logistics network.
  - Higher volume of orders, reducing per-order shipping cost.

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

```
WITH final AS (
    SELECT
        c.customer_state AS state,
        SUM(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,
DAY)) AS total_delivery_time_days,
        COUNT(DISTINCT o.order_id) AS no_of_orders
    FROM `Target.customers` c
    INNER JOIN `Target.orders` o ON c.customer_id = o.customer_id

    GROUP BY c.customer_state
),
average_data as(
SELECT
    state,
    ROUND(total_delivery_time_days / no_of_orders, 2) AS avg_delivery_time
FROM final
ORDER BY avg_delivery_time
)


(select state,
       avg_delivery_time,
       row_number()over(order by avg_delivery_time desc) as ranking,
       "Highest" as type
from average_data
limit 5)
```

```
union all
(select state,
       avg_delivery_time,
       row_number()over(order by avg_delivery_time) as ranking,
       "lowest" as type
from average_data
limit 5)
```

| state | avg_delivery_time | ranking | type |
|---|---|---|---|
| SP | 8.05 | 1 | lowest |
| PR | 11.25 | 2 | lowest |
| MG | 11.27 | 3 | lowest |
| DF | 12.16 | 4 | lowest |
| SC | 14.12 | 5 | lowest |
| AP | 26.34 | 1 | Highest |
| RR | 25.83 | 2 | Highest |
| AM | 25.46 | 3 | Highest |
| AL | 23.11 | 4 | Highest |
| PA | 22.62 | 5 | Highest |

**1. Fastest Delivery States**

- **These states have the shortest average delivery times.**
- **Possible reasons:**
  - **Efficient local logistics and faster shipping routes.**
  - **Proximity to warehouses or distribution centers.**
  - **Higher order volumes, leading to optimized delivery schedules.**

**2. Slowest Delivery States**

- **These states have the longest average delivery times.**
- **Possible reasons:**
  - **Remote locations with fewer transport options.**
  - **Poor infrastructure, leading to delivery delays.**
  - **Unoptimized logistics routes causing inefficiencies.**

### 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.

```
with final as
(select c.customer_state as state,
       count(distinct o.order_id) as no_of_orders,
     sum(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))
as total_actual_delivery_days,
       sum(date_diff(order_estimated_delivery_date,o.order_purchase_timestamp,day))
as total_estimated_delivery_days
from `Target.orders` o
inner join `Target.customers` c
ON o.customer_id = c.customer_id
group by state),

avg_data as
(select state,
       no_of_orders,
       round(total_actual_delivery_days/no_of_orders,2) as avg_actual_delivery,
       round(total_estimated_delivery_days/no_of_orders,2) as
avg_estimated_delivery
from final
order by 3,4)


select state,
       no_of_orders,
       round(avg_estimated_delivery-avg_actual_delivery,2) as fast_deliveries
from avg_data
order by fast_deliveries desc
limit 5
```

| state | no_of_orders | fast_deliveries |
|---|---:|---:|
| AC | 81 | 20.39 |
| RR | 46 | 20.34 |
| RO | 253 | 20.24 |
| AP | 68 | 19.37 |
| AM | 148 | 19.3 |

**1. Top 5 Fastest Delivery States**

- **These states have a positive "fast_deliveries" value, meaning orders are delivered earlier than the estimated date.**
- **Possible reasons:**
    - **Highly optimized logistics & supply chain.**
    - **Presence of local warehouses.**
    - **Strong transportation networks (e.g., highways, airports, courier services).**

# 6.Analysis based on the payments:

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

```
SELECT extract (month from order_purchase_timestamp) as month,
       count(o.order_id) as no_of_orders,
       p.payment_type as payment_mode

from `Target.orders` o
inner join `Target.payments` p
ON o.order_id = p.order_id
group by month,payment_mode
order by month
```

| month | no_of_orders | payment_mode |
|---:|---:|---|
| 1 | 6103 | credit_card |
| 1 | 1715 | UPI |
| 1 | 477 | voucher |
| 1 | 118 | debit_card |
| 2 | 1723 | UPI |
| 2 | 6609 | credit_card |
| 2 | 424 | voucher |
| 2 | 82 | debit_card |
| 3 | 7707 | credit_card |
| 3 | 1942 | UPI |
| 3 | 109 | debit_card |
| 3 | 591 | voucher |
| 4 | 572 | voucher |
| 4 | 7301 | credit_card |

## In-sights :

**1.If a particular payment mode is dominant, it may indicate customer trust in that method.**

**2.Business can offer targeted promotions (e.g., cashback on credit card payments).**

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

```
SELECT payment_installments AS installments,
       COUNT(distinct order_id) AS num_orders,
FROM `Target.payments`
WHERE payment_installments >= 1
GROUP BY payment_installments
ORDER BY num_orders DESC
```

| installments | num_orders |
|---:|---:|
| 1 | 49060 |
| 2 | 12389 |
| 3 | 10443 |
| 4 | 7088 |
| 10 | 5315 |
| 5 | 5234 |
| 8 | 4253 |
| 6 | 3916 |
| 7 | 1623 |
| 9 | 644 |
| 12 | 133 |
| 15 | 74 |
| 18 | 27 |
| 11 | 23 |
| 24 | 18 |
| 20 | 17 |
| 13 | 16 |
| 14 | 15 |
| 17 | 8 |
| 16 | 5 |
| 21 | 3 |

| | |
|---|---|
| 22 | 1 |
| 23 | 1 |

**Payment Behavior:**

- **You can identify which installment plans are most popular among customers.**
- **Helps to understand how many customers prefer to pay in installments rather than upfront.**