

# Business Case: E-commerce brand

**Project Description:** This project centres on analysing the operations of a globally recognized American retailer, focusing on its presence in Brazil. With a dataset encompassing 100,000 orders between 2016 and 2018, it aims to uncover insights into order processing, pricing strategies, payment and shipping efficiency, customer demographics, and satisfaction levels. The findings will inform strategic decisions for the company's operations in Brazil.

## Tech-Stack Used:

1. MySQL Workbench 8.0
2. Tableau Public (Visualization)

**Name of Database:** target

## Section 1: Initial data exploration

→ Exploratory analysis steps like checking the structure & characteristics of the dataset:

**Query 1.1: What are the datatypes of the column across all the tables**

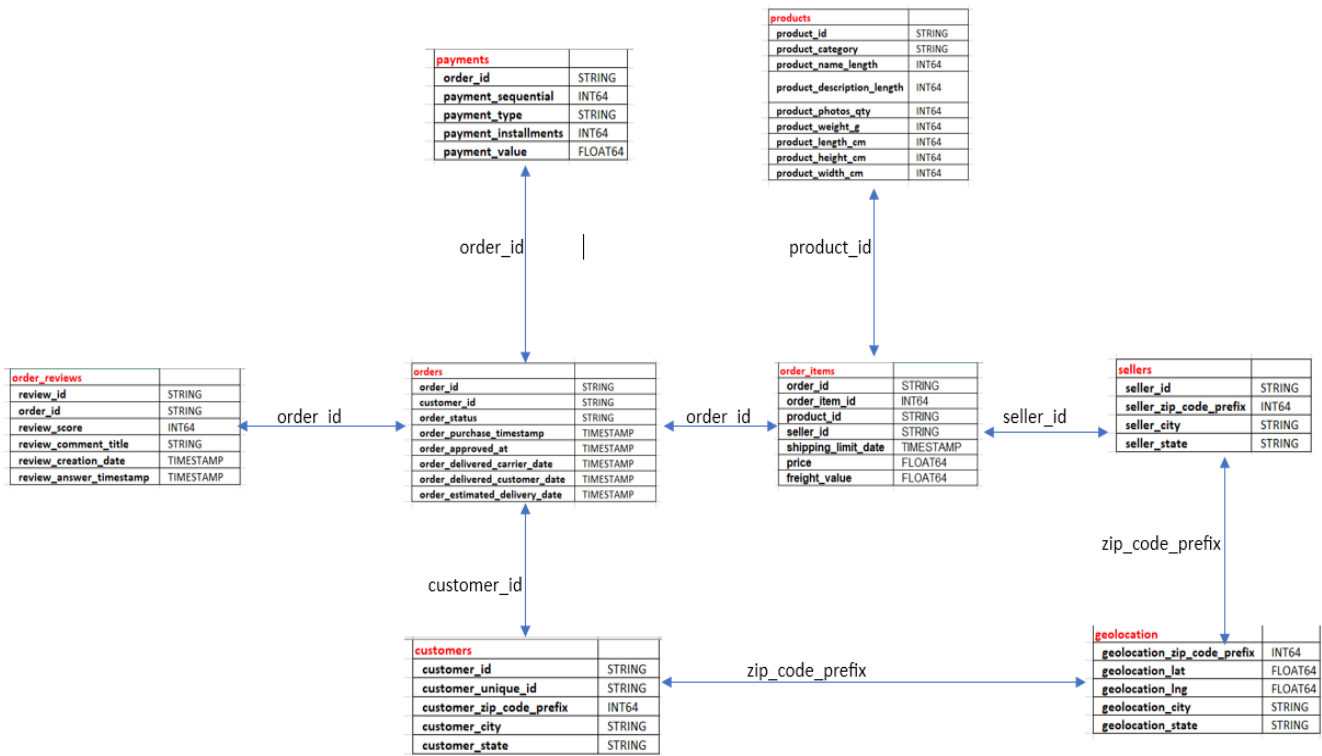
**Sample Query For column: 'customers'**

```
describe target.customers
```

| column_name              | data_type |
|--------------------------|-----------|
| customer_id              | STRING    |
| customer_unique_id       | STRING    |
| customer_zip_code_pre... | INT64     |
| customer_city            | STRING    |
| customer_state           | STRING    |

**Note:** Similar query can be used for understanding data of other columns in the dataset.

## Dataset Schema:



**Query 1.2 Get the time range between which the orders were placed.**

```
SELECT
  MIN(order_purchase_timestamp) AS first_order_date,
  MAX(order_purchase_timestamp) AS last_order_date,
  DATEDIFF(MAX(order_purchase_timestamp),MIN(order_purchase_timestamp)) AS time_range_days,
  DATEDIFF(MAX(order_purchase_timestamp), MIN(order_purchase_timestamp)) / 30 AS time_range_months
FROM
  target.orders
```

**Result:**

|   | first_order_date    | last_order_date     | time_range_days | time_range_months |
|---|---------------------|---------------------|-----------------|-------------------|
| ► | 2016-09-04 21:15:19 | 2018-10-17 17:30:18 | 773             | 25.7667           |

**Query 1.3: In how many cities and states the company operates their business?**

```
SELECT
  COUNT(DISTINCT c.customer_city) AS city_count,
  COUNT(DISTINCT c.customer_state) AS state_count
FROM
  target.customers c
JOIN
  target.orders o ON c.customer_id = o.customer_id
```

**Result:**

|   | city_count | state_count |
|---|------------|-------------|
| ► | 4119       | 27          |

**Query 1.4: How many products and product categories does company focusses on?**

```
SELECT
  COUNT(DISTINCT product_id) AS Products_count,
  COUNT(DISTINCT product_category) AS Product_categories_count
FROM
  target.products
```

**Result:**

|   | Products_count | Product_categories_count |
|---|----------------|--------------------------|
| ► | 32951          | 73                       |

**Query 1.5: What is the total number of customers in Brazil**

```
SELECT
  COUNT(DISTINCT customer_id) as total_customers
FROM target.customers
```

**Result:**

|   | total_customers |
|---|-----------------|
| ► | 99441           |

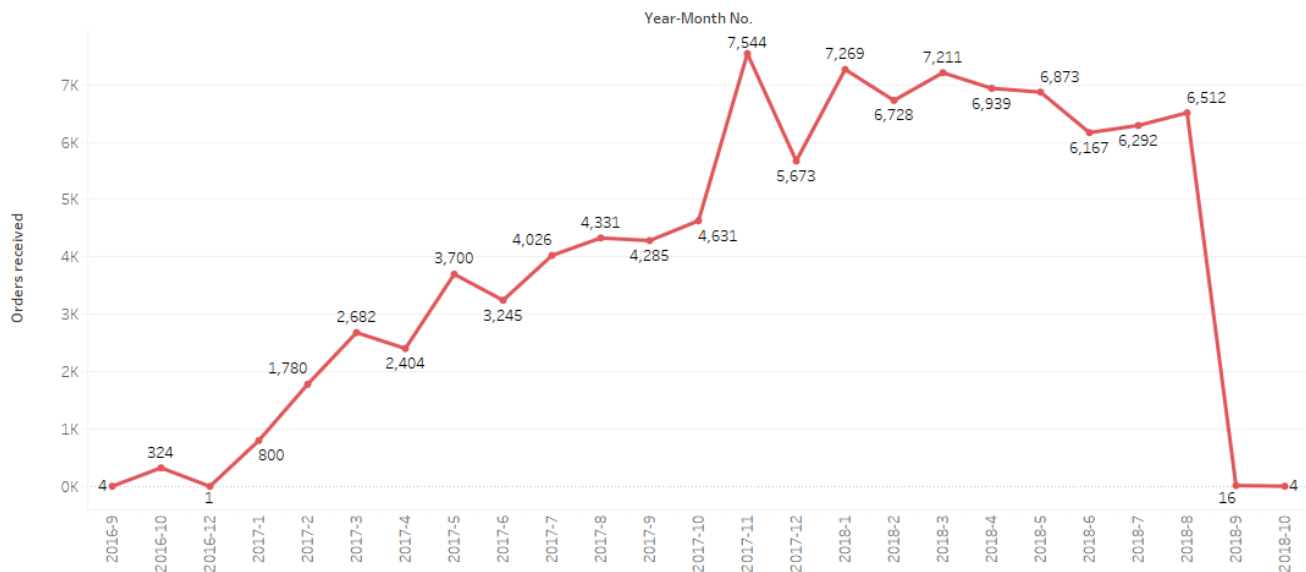
## Section-2: In-depth Exploration

**Query 2.1: Can we see some kind of monthly seasonality in terms of the no. of orders being placed?**

```
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS Year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS Month,
  COUNT(order_id) AS Orders_received
FROM
  target.orders
GROUP BY Year , Month
ORDER BY Year , Month
```

**Result:**

|   | Year | Month | Orders_received |
|---|------|-------|-----------------|
| ▶ | 2016 | 9     | 4               |
|   | 2016 | 10    | 324             |
|   | 2016 | 12    | 1               |
|   | 2017 | 1     | 800             |
|   | 2017 | 2     | 1780            |
|   | 2017 | 3     | 2682            |
|   | 2017 | 4     | 2404            |
|   | 2017 | 5     | 3700            |
|   | 2017 | 6     | 3245            |
|   | 2017 | 7     | 4026            |
|   | 2017 | 8     | 4331            |
|   | 2017 | 9     | 4285            |
|   | 2017 | 10    | 4631            |
|   | 2017 | 11    | 7544            |
|   | 2017 | 12    | 5673            |
|   | 2018 | 1     | 7269            |
|   | 2018 | 2     | 6728            |
|   | 2018 | 3     | 7211            |
|   | 2018 | 4     | 6939            |
|   | 2018 | 5     | 6873            |
|   | 2018 | 6     | 6167            |
|   | 2018 | 7     | 6292            |
|   | 2018 | 8     | 6512            |
|   | 2018 | 9     | 16              |
|   | 2018 | 10    | 4               |

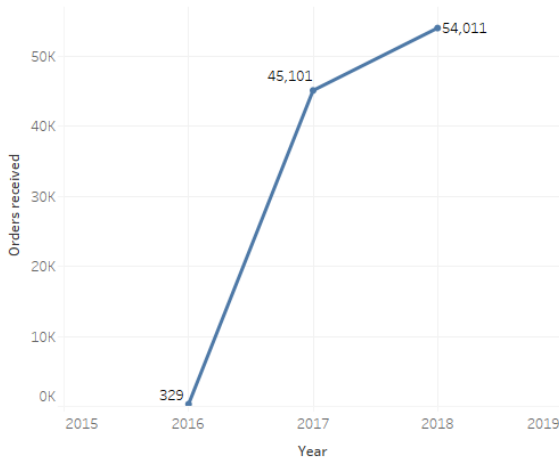


**Query 2: Is there a growing trend in the no. of orders placed over the past years?**

```
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS Year,
  COUNT(order_id) AS Orders_received
FROM
  target.orders
GROUP BY Year
ORDER BY Year
```

**Result:**

|   | Year | Orders_received |
|---|------|-----------------|
| ► | 2016 | 329             |
|   | 2017 | 45101           |
|   | 2018 | 54011           |

**Query 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**

**SELECT**

*SUM(CASE WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 0 AND 6 THEN 1 ELSE 0 END) AS Dawn\_orders,*

*SUM(CASE WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 7 AND 12 THEN 1 ELSE 0 END) AS Mornings\_orders,*

*SUM(CASE WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 13 AND 18 THEN 1 ELSE 0 END) AS Afternoon\_orders,*

*SUM(CASE WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 19 AND 23 THEN 1 ELSE 0 END) AS Night\_orders*

**FROM**

*target.orders*

**Result**

|   | Dawn_orders | Mornings_orders | Afternoon_orders | Night_orders |
|---|-------------|-----------------|------------------|--------------|
| ► | 5242        | 27733           | 38135            | 28331        |

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

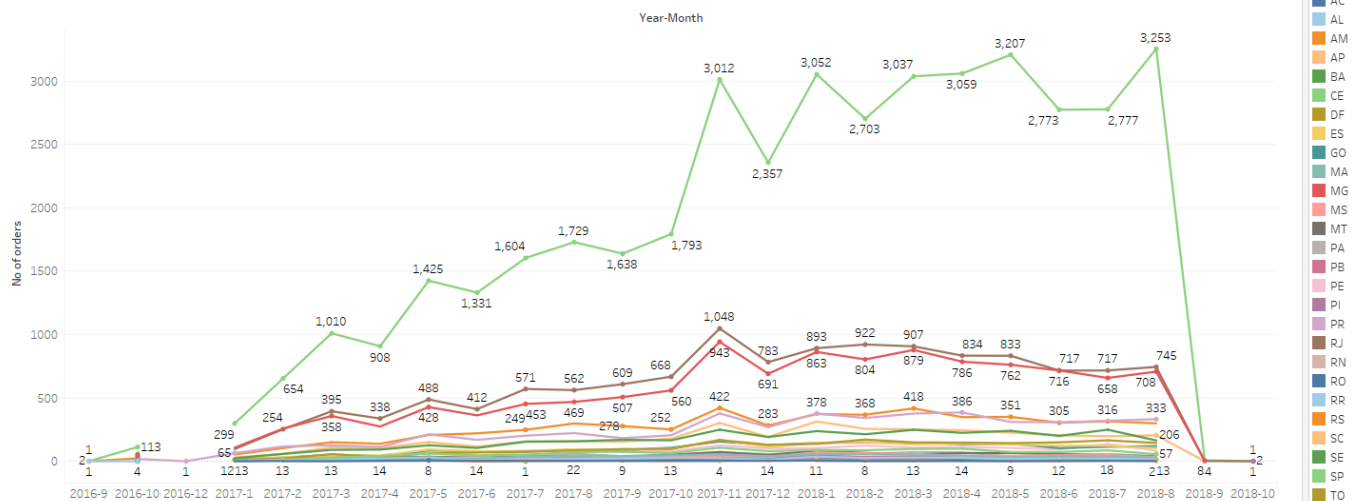
**Query 3.1: Get the month-on-month no. of orders placed in each state.**

```
SELECT
    customer_state AS State,
    EXTRACT(YEAR FROM order_purchase_timestamp) AS Year,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS Month,
    COUNT(order_id) AS No_of_orders
FROM
    target.orders o
JOIN
    target.customers c ON o.customer_id = c.customer_id
GROUP BY customer_state , Year , Month
ORDER BY State , Year , Month
```

**Result:**

| States | Year | Month | No_of_orders |
|--------|------|-------|--------------|
| AC     | 2017 | 1     | 2            |
| AC     | 2017 | 2     | 3            |
| AC     | 2017 | 3     | 2            |
| AC     | 2017 | 4     | 5            |
| AC     | 2017 | 5     | 8            |
| AC     | 2017 | 6     | 4            |
| AC     | 2017 | 7     | 5            |
| AC     | 2017 | 8     | 4            |
| AC     | 2017 | 9     | 5            |
| AC     | 2017 | 10    | 6            |
| AL     | 2016 | 10    | 2            |
| AL     | 2017 | 1     | 2            |
| AL     | 2017 | 2     | 12           |
| AL     | 2017 | 3     | 10           |
| AL     | 2017 | 4     | 23           |
| AL     | 2017 | 5     | 27           |
| AL     | 2017 | 6     | 10           |
| AL     | 2017 | 7     | 17           |
| AI     | 2017 | 8     | 18           |
| AM     | 2017 | 2     | 8            |
| AM     | 2017 | 3     | 5            |
| AM     | 2017 | 4     | 13           |
| AM     | 2017 | 5     | 10           |
| AM     | 2017 | 6     | 1            |
| AM     | 2017 | 7     | 5            |
| AM     | 2017 | 8     | 5            |
| AM     | 2017 | 9     | 9            |
| AM     | 2017 | 10    | 3            |
| AM     | 2017 | 11    | 10           |

# of Orders placed on month-to-month basis for each state



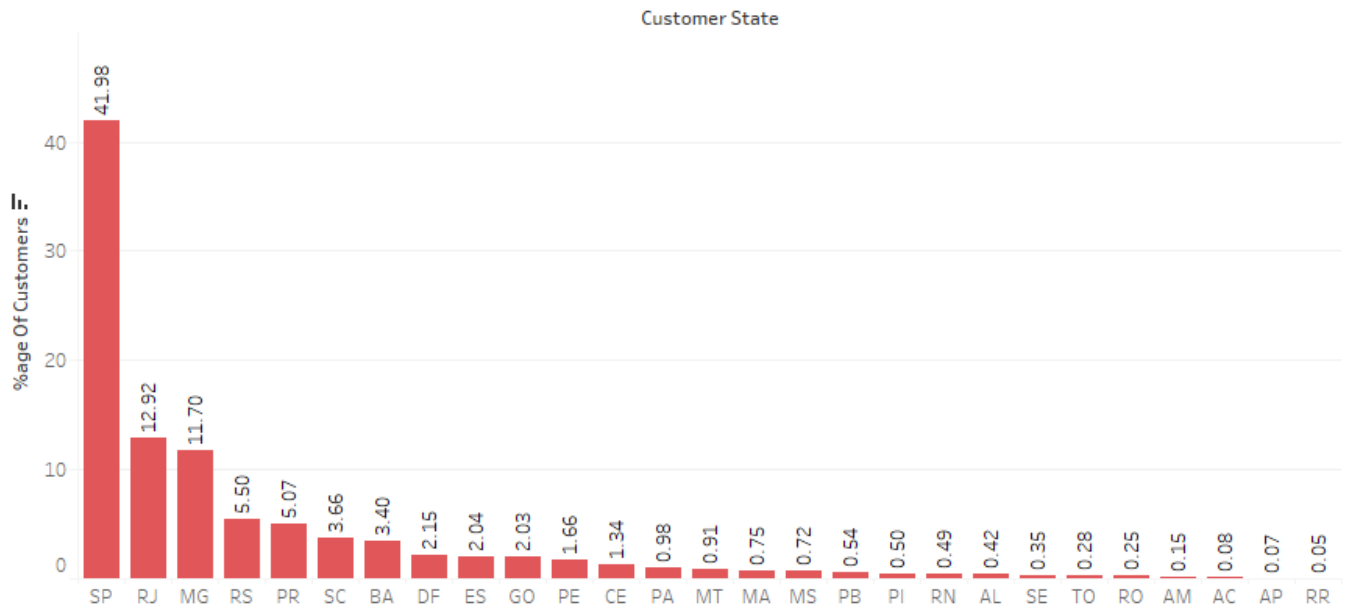
**Query 2: How are the customers distributed across all the states?**

```
SELECT
    customer_state,
    ROUND(COUNT(customer_state) * 100 / (SELECT COUNT(*) FROM target.customers),2) AS '%age of customers'
FROM
    target.customers
GROUP BY customer_state
ORDER BY 2 DESC
```

## Result:

| customer_state | %age of customers |
|----------------|-------------------|
| SP             | 41.98             |
| RJ             | 12.92             |
| MG             | 11.70             |
| RS             | 5.50              |
| PR             | 5.07              |
| SC             | 3.66              |
| BA             | 3.40              |
| DF             | 2.15              |
| ES             | 2.04              |
| GO             | 2.03              |
| PE             | 1.66              |
| CE             | 1.34              |

|    |      |
|----|------|
| PA | 0.98 |
| MT | 0.91 |
| MA | 0.75 |
| MS | 0.72 |
| PB | 0.54 |
| PI | 0.50 |
| RN | 0.49 |
| AL | 0.42 |
| SE | 0.35 |
| TO | 0.28 |
| RO | 0.25 |
| AM | 0.15 |
| AC | 0.08 |
| AP | 0.07 |
| RR | 0.05 |



#### Section-4: Impact on Economy:

→Analysing the money movement by e-commerce by looking at order prices, freight, and others.

**Query 4.1: Estimation of % increase in cost of orders from year 2017 to 2018 (include months between Jan-Aug only).**

```
SELECT
    Year, x.curr_payment,
    LAG(x.curr_payment,1,0) OVER () as prev_payment,
    ROUND(IFNULL((x.curr_payment - LAG(curr_payment,1,0) OVER () ) *100/ (LAG(x.curr_payment,1,0) OVER
    ()),0),2) as 'percent_rise_payment'
FROM
    (SELECT
        distinct YEAR(o.order_purchase_timestamp) as Year,
        round(SUM(p.payment_value) OVER (PARTITION BY YEAR(o.order_purchase_timestamp)),2) as
curr_payment
    FROM target.orders o
    JOIN target.payments p ON o.order_id=p.order_id
    WHERE
        (YEAR(o.order_purchase_timestamp) between 2017 AND 2018) AND
        (MONTH(o.order_purchase_timestamp) between 1 AND 8)) x
```

**Result:**

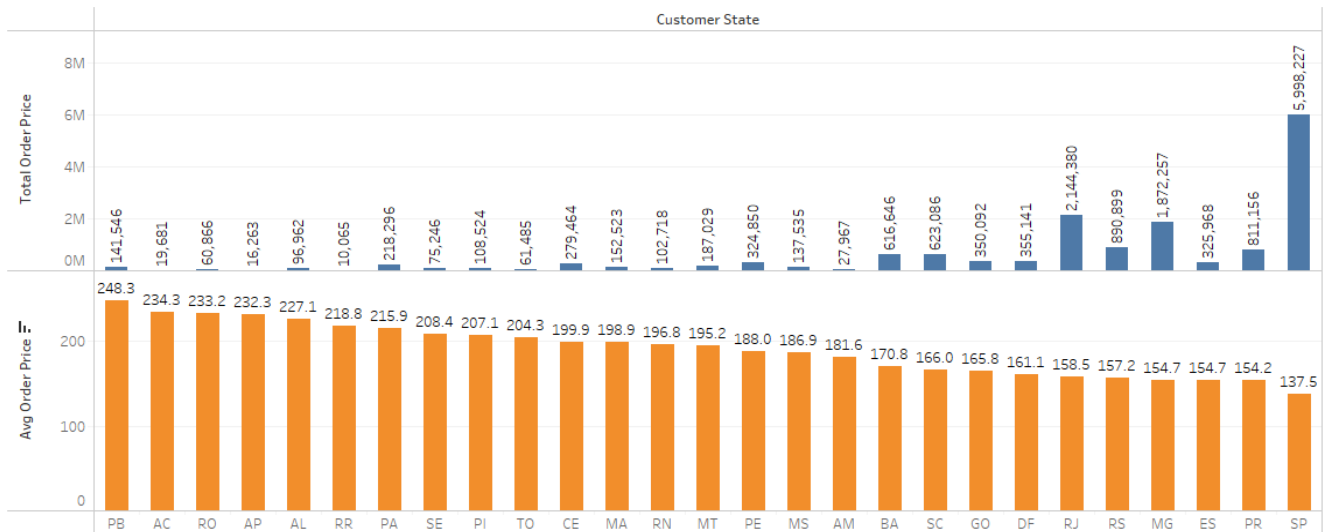
|   | Year | curr_payment | prev_payment | percent_rise_payment |
|---|------|--------------|--------------|----------------------|
| ▶ | 2017 | 3669022.12   | 0            | 0                    |
|   | 2018 | 8694733.84   | 3669022.12   | 136.98               |

**Query 4.2: Calculate the Total & Average value of order price for each state.**

```
SELECT
    c.customer_state,
    ROUND(SUM(p.payment_value), 2) AS total_order_price,
    ROUND(AVG(p.payment_value), 2) AS avg_order_price
FROM target.customers c
JOIN
    target.orders o ON c.customer_id = o.customer_id
JOIN
    target.payments p ON o.order_id = p.order_id
GROUP BY c.customer_state
ORDER BY 2 DESC , 3 DESC
```

|   | customer_state | total_order_price | avg_order_price |
|---|----------------|-------------------|-----------------|
| ▶ | SP             | 5998226.96        | 137.5           |
|   | RJ             | 2144379.69        | 158.53          |
|   | MG             | 1872257.26        | 154.71          |
|   | RS             | 890898.54         | 157.18          |
|   | PR             | 811156.38         | 154.15          |
|   | SC             | 623086.43         | 165.98          |
|   | BA             | 616645.82         | 170.82          |
|   | DF             | 355141.08         | 161.13          |
|   | GO             | 350092.31         | 165.76          |
|   | ES             | 325967.55         | 154.71          |
|   | PE             | 324850.44         | 187.99          |
|   | CE             | 279464.03         | 199.9           |
|   | PA             | 218295.85         | 215.92          |

|  |    |           |        |
|--|----|-----------|--------|
|  | MT | 187029.29 | 195.23 |
|  | MA | 152523.02 | 198.86 |
|  | PB | 141545.72 | 248.33 |
|  | MS | 137534.84 | 186.87 |
|  | PI | 108523.97 | 207.11 |
|  | RN | 102718.13 | 196.78 |
|  | AL | 96962.06  | 227.08 |
|  | SE | 75246.25  | 208.44 |
|  | TO | 61485.33  | 204.27 |
|  | RO | 60866.2   | 233.2  |
|  | AM | 27966.93  | 181.6  |
|  | AC | 19680.62  | 234.29 |
|  | AP | 16262.8   | 232.33 |
|  | RR | 10064.62  | 218.8  |



**Query 4.3: Calculate the Total & Average value of order freight for each state.**

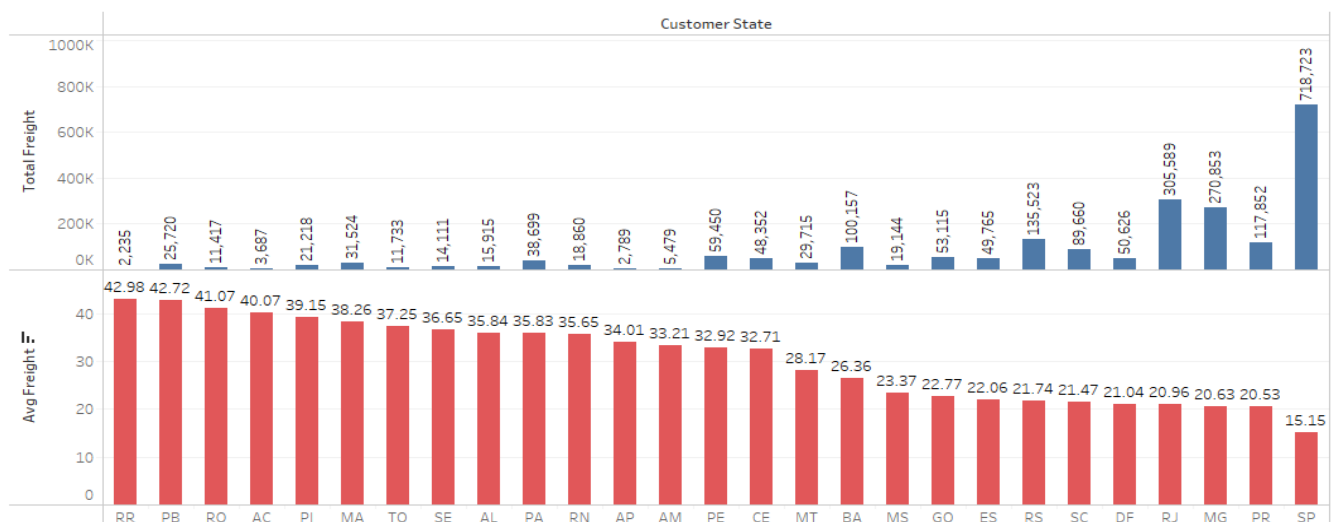
```

SELECT
  c.customer_state,
  ROUND(SUM(oi.freight_value), 2) AS total_freight,
  ROUND(AVG(oi.freight_value), 2) AS avg_freight
FROM target.customers c
JOIN target.orders o ON c.customer_id = o.customer_id
JOIN target.order_items oi ON oi.order_id = o.order_id
GROUP BY c.customer_state
ORDER BY 2 DESC, 3 DESC

```

|   | customer_state | total_freight | avg_freight |
|---|----------------|---------------|-------------|
| ▶ | SP             | 718723.07     | 15.15       |
|   | RJ             | 305589.31     | 20.96       |
|   | MG             | 270853.46     | 20.63       |
|   | RS             | 135522.74     | 21.74       |
|   | PR             | 117851.68     | 20.53       |
|   | BA             | 100156.68     | 26.36       |
|   | SC             | 89660.26      | 21.47       |
|   | PE             | 59449.66      | 32.92       |
|   | GO             | 53114.98      | 22.77       |
|   | DF             | 50625.5       | 21.04       |
|   | ES             | 49764.6       | 22.06       |
|   | CE             | 48351.59      | 32.71       |
|   | PA             | 38699.3       | 35.83       |

|  |    |          |       |
|--|----|----------|-------|
|  | MA | 31523.77 | 38.26 |
|  | MT | 29715.43 | 28.17 |
|  | PB | 25719.73 | 42.72 |
|  | PI | 21218.2  | 39.15 |
|  | MS | 19144.03 | 23.37 |
|  | RN | 18860.1  | 35.65 |
|  | AL | 15914.59 | 35.84 |
|  | SE | 14111.47 | 36.65 |
|  | TO | 11732.68 | 37.25 |
|  | RO | 11417.38 | 41.07 |
|  | AM | 5478.89  | 33.21 |
|  | AC | 3686.75  | 40.07 |
|  | AP | 2788.5   | 34.01 |
|  | RR | 2235.19  | 42.98 |





## Section 5: Analysis based on sales, freight, and delivery time.

### Query 5.1: Find the Estimated and Actual time of delivery of an order in days

```
SELECT
    TIMESTAMPDIFF(DAY, order_purchase_timestamp, order_estimated_delivery_date) AS Estimated_time,
    TIMESTAMPDIFF(DAY, order_purchase_timestamp, order_delivered_customer_date) AS Actual_time
FROM
    target.orders
WHERE
    order_status = 'delivered'
ORDER BY 2 DESC
```

#### Result:

|   | Estimated_time | Actual_time |   | Estimated_time | Actual_time |
|---|----------------|-------------|---|----------------|-------------|
| ► | 28             | 209         | ► | 155            | 20          |
|   | 19             | 208         |   | 149            | 3           |
|   | 30             | 195         |   | 146            | 6           |
|   | 39             | 194         |   | 140            | 16          |
|   | 32             | 194         |   | 116            | 7           |
|   | 28             | 194         |   | 109            | 54          |
|   | 15             | 191         |   | 101            | 63          |
|   | 22             | 189         |   | 99             | 47          |
|   | 28             | 188         |   | 97             | 28          |
|   | 25             | 187         |   | 96             | 55          |
|   | 42             | 187         |   | 95             | 12          |

### Query 5.2: Find out the top 5 states with the highest average freight value.

```
SELECT
    customer_state as Top_5_states,
    ROUND(AVG(oi.freight_value), 2) avg_freight_value
FROM
    target.customers c
JOIN
    target.orders o ON c.customer_id = o.customer_id
JOIN
    target.order_items oi ON o.order_id = oi.order_id
GROUP BY
    customer_state
ORDER BY avg_freight_value DESC
LIMIT 5
```

#### Result:

|   | Top_5_states | avg_freight_value |
|---|--------------|-------------------|
| ► | RR           | 42.98             |
|   | PB           | 42.72             |
|   | RO           | 41.07             |
|   | AC           | 40.07             |
|   | PI           | 39.15             |

**Query 5.3: Find out the top 5 states with the lowest average freight value.**

```
SELECT
    customer_state AS Top_5_states,
    ROUND(AVG(oi.freight_value), 2) avg_freight_value
FROM
    target.customers c
JOIN
    target.orders o ON c.customer_id = o.customer_id
JOIN
    target.order_items oi ON o.order_id = oi.order_id
GROUP BY customer_state
ORDER BY avg_freight_value ASC
LIMIT 5
```

|   | Top_5_states | avg_freight_value |
|---|--------------|-------------------|
| ▶ | SP           | 15.15             |
|   | PR           | 20.53             |
|   | MG           | 20.63             |
|   | RJ           | 20.96             |
|   | DF           | 21.04             |

**Query 5.4: Find out the top 5 states with the highest average delivery time.**

```
SELECT
    c.customer_state,
    ROUND(AVG(TIMESTAMPDIFF(DAY, o.order_purchase_timestamp, o.order_delivered_customer_date)),2)
AS avg_delivery_time_days
FROM target.orders o
JOIN
    target.customers c ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY 2 DESC
LIMIT 5
```

|   | customer_state | avg_delivery_time_days |
|---|----------------|------------------------|
| ▶ | RR             | 28.98                  |
|   | AP             | 26.73                  |
|   | AM             | 25.99                  |
|   | AL             | 24.04                  |
|   | PA             | 23.32                  |

**Query 5.5: Find out the top 5 states with the lowest average delivery time.**

```
SELECT
    c.customer_state,
    ROUND(AVG(TIMESTAMPDIFF(DAY, o.order_purchase_timestamp, o.order_delivered_customer_date)),2)
AS avg_delivery_time_days
FROM target.orders o
JOIN
    target.customers c ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY 2 asc
```

|   | customer_state | avg_delivery_time_days |
|---|----------------|------------------------|
| ▶ | SP             | 8.30                   |
|   | PR             | 11.53                  |
|   | MG             | 11.54                  |
|   | DF             | 12.51                  |
|   | SC             | 14.48                  |

**Query 5.6: Finding the top 5 states where the order delivery is fast as compared to the estimated date of delivery. We used 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(TIMESTAMPDIFF(DAY,o.order_purchase_timestamp,o.order_delivered_customer_date)-
    timestampdiff(day,o.order_purchase_timestamp,o.order_estimated_delivery_date)),2) as
  diff_actual_estimated_days
FROM
  target.orders o
JOIN
  target.customers c ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY 2 asc
LIMIT 5

```

|   | customer_state | diff_actual_estimated_days |
|---|----------------|----------------------------|
| ▶ | AC             | -20.09                     |
|   | RO             | -19.47                     |
|   | AP             | -19.13                     |
|   | AM             | -18.94                     |
|   | RR             | -16.66                     |

## Section 6: Payment analysis

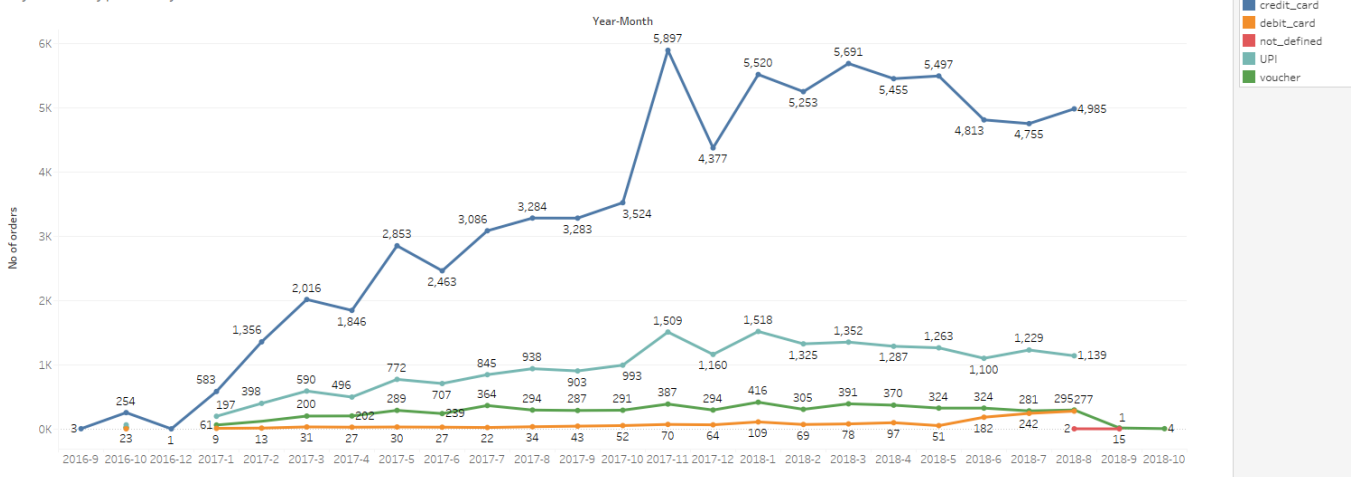
**Query 6.1: Find the month-on-month no. of orders placed using different payment types.**

```
SELECT
    p.payment_type AS Payment_Type,
    YEAR(o.order_purchase_timestamp) AS Year,
    MONTH(o.order_purchase_timestamp) AS Month,
    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 payment_type , Year , Month
ORDER BY 1 , 2 , 3
```

**Result:**

|   | Payment_Type | Year | Month | No_of_orders |
|---|--------------|------|-------|--------------|
| ▶ | credit_card  | 2016 | 9     | 3            |
|   | credit_card  | 2016 | 10    | 254          |
|   | credit_card  | 2016 | 12    | 1            |
|   | credit_card  | 2017 | 1     | 583          |
|   | credit_card  | 2017 | 2     | 1356         |
|   | credit_card  | 2017 | 3     | 2016         |
|   | credit_card  | 2017 | 4     | 1846         |
|   | credit_card  | 2017 | 5     | 2853         |
|   | credit_card  | 2017 | 6     | 2463         |
|   | credit_card  | 2017 | 7     | 3086         |
|   | debit_card   | 2017 | 8     | 34           |
|   | debit_card   | 2017 | 9     | 43           |
|   | debit_card   | 2017 | 10    | 52           |
|   | debit_card   | 2017 | 11    | 70           |
|   | debit_card   | 2017 | 12    | 64           |
|   | debit_card   | 2018 | 1     | 109          |
|   | debit_card   | 2018 | 2     | 69           |
|   | debit_card   | 2018 | 3     | 78           |
|   | debit_card   | 2018 | 4     | 97           |
|   | debit_card   | 2018 | 5     | 51           |
|   | debit_card   | 2018 | 6     | 182          |

Payment\_Type Analysis

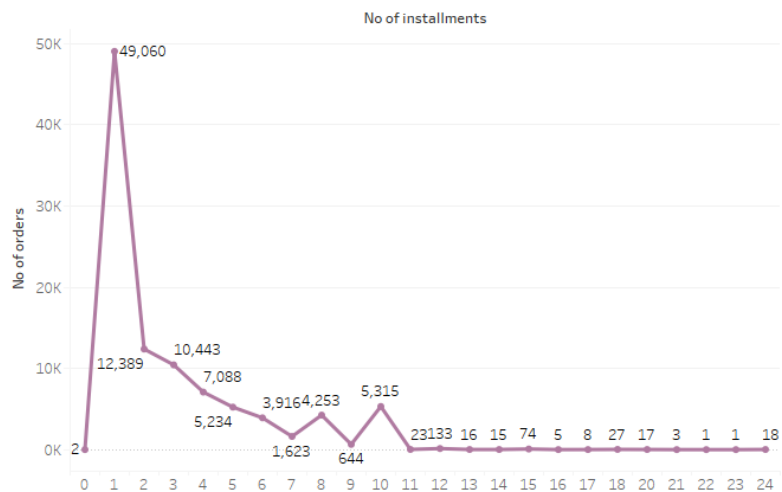


**Query 6.2: Find the number of orders placed based on the payment instalments that have been paid.**

```
SELECT
    payment_installments AS No_of_installments,
    COUNT(DISTINCT order_id) AS No_of_orders
FROM
    target.payments
GROUP BY payment_installments
ORDER BY 1
```

**Result:**

|     | No_of_installments | No_of_orders |
|-----|--------------------|--------------|
| ▶ 0 |                    | 2            |
| 1   |                    | 49060        |
| 2   |                    | 12389        |
| 3   |                    | 10443        |
| 4   |                    | 7088         |
| 5   |                    | 5234         |
| 6   |                    | 3916         |
| 7   |                    | 1623         |
| 8   |                    | 4253         |
| 9   |                    | 644          |
| 10  |                    | 5315         |
| 11  |                    | 23           |



**Query 6.3: What is total revenue generated from all payments and percentage distribution of revenue across various payment types**

```

select
    x.payment_type,
    x.Payment_type_revenue,
    round(sum(x.Payment_type_revenue)over(),2) as Total_Revenue,
    round((x.Payment_type_revenue)*100/sum(x.Payment_type_revenue)over(),2) as "Revenue_percentage"
from
    (select
        payment_type,
        round(sum(p.payment_value),2) as Payment_type_revenue
    from target.payments p
    join target.orders o
    on o.order_id=p.order_id
    group by p.payment_type
    )x
order by 4 desc

```

|   | payment_type | Payment_type_revenue | Total_Revenue | Revenue_percentage |
|---|--------------|----------------------|---------------|--------------------|
| ▶ | credit_card  | 12542084.19          | 16008872.12   | 78.34              |
|   | UPI          | 2869361.27           | 16008872.12   | 17.92              |
|   | voucher      | 379436.87            | 16008872.12   | 2.37               |
|   | debit_card   | 217989.79            | 16008872.12   | 1.36               |
|   | not_defined  | 0                    | 16008872.12   | 0                  |

## Insights:

1. The dataset spans a duration of 773 days, equivalent to 26 months, starting from the first order placed on '2016-09-04' and concluding with the last order placed on '2018-10-17'
2. The company conducts its business operations in 4,119 cities across 27 states.
3. The company offers 32,951 distinct products across 73 different product categories.
4. There is a clear rising trend in the business in Brazil.
5. Brazilian customers exhibit a tendency to place their orders primarily in the afternoon (13:00 hr-18:00 hrs), followed by the night (19:00-23:00 hrs) and mornings (00-06 hrs).
6. The company's largest customer base in Brazil is in São Paulo (SP) at 41.98%, followed by Rio de Janeiro (RJ) at 12.92%.
7. There is a significant spike in the number of orders, with a 137% increase observed from 2017 to 2018.
8. Top 5-States with the highest average freight value are RR, PB, RO AC, PI.
9. Top 5-States with the lowest average freight value are SP, PR, MG, RJ, DF.
10. Top 5-States with the highest average delivery time are RR, AP, AM, AL, PA.
11. Top 5-States with the lowest average delivery time are SP, PR, MG, DF, SC.
12. Top 5-States where order delivery is very fast are AC RO, AP, AM, RR.
13. Brazilian customers exhibit a preference for the payment mode of Credit Card, accounting for 78.34%, followed by UPI (Unified Payments Interface) at 17.92%.
14. Brazilian customers prefer making payments in a single instalment (i.e., a one-time payment), followed by either two or three instalments.
15. The total revenue generated from all modes of payment amounts to 16,008,872.12

## Recommendations:

1. Company should continue to promote credit card payments, given the high preference among Brazilian customers.
2. Consider introducing attractive incentives for customers opting for single installment payments, or explore partnerships with financial institutions for installment plans.
3. Consider special promotions or discounts during peak order placement times, especially in the afternoon and night.
4. Identify popular products and categories in Brazil and consider introducing related or complementary items.
5. Focus on improving delivery times in states with higher average delivery times to enhance customer satisfaction.
6. Consider loyalty programs or exclusive offers for customers in São Paulo and Rio de Janeiro, the largest customer bases.
7. Collaborate with local businesses or logistics partners to further enhance delivery efficiency and reduce average delivery times.
8. Identify cities or states with untapped potential for business growth and consider expanding operations in those areas.