

PROJECT NUMBER 1

**BAHADURSHA A V L SAINADH**

SCALER-JULY 2022 BEGINNER BATCH

**A sql project on brazil’s target STORE dataset**



submitted on: 18th january, 2023

last date to submit: 18th january, 2023

SCALER DATA SCIENCE, ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCALER NEOVARSITY / WOOLF UNIVERSITY

TABLE OF CONTENTS

[CHAPTER 1: INTRODUCTION 1](#_Toc124962598)

[1.1 INTRODUCTION TO TARGET CORPORATION 1](#_Toc124962599)

[1.2 business case details 1](#_Toc124962600)

[1.2.1 Features of customers.csv 1](#_Toc124962601)

[1.2.2 Features of geolocations.csv 2](#_Toc124962602)

[1.2.3 Features of orders.csv 2](#_Toc124962603)

[1.2.4 Features of order\_items.csv 2](#_Toc124962604)

[1.2.5 Features of order\_reviews.csv 2](#_Toc124962605)

[1.2.6 Features of payments.csv 3](#_Toc124962606)

[1.2.7 Features of products.csv 3](#_Toc124962607)

[1.2.8 Features of sellers.csv 3](#_Toc124962608)

[1.3 Entity Relationship Diagram 3](#_Toc124962609)

[CHAPTER 2: IMPORTING THE DATASET AND PRELIMINARY EXPLORATORY ANALYSIS……………………………………………………………………………………4](#_Toc124962610)

[2.1 Importing the dataset into google’s bigquery sandbox 4](#_Toc124962611)

[2.1.1 Why BigQuery Sandbox? 4](#_Toc124962612)

[2.2 Datatypes of Every column in all tables 4](#_Toc124962613)

[2.3 Finding primary keys of each and every table 6](#_Toc124962614)

[2.4 Time period of given data 7](#_Toc124962615)

[2.5 Distinct Cities and States of Customers who ordered in 772 days’ time period 8](#_Toc124962616)

[CHAPTER 3: IN-DEPTH EXPLORATION 9](#_Toc124962617)

[3.1 Complete Scenario of E-commerce trend (sales over TIME) 9](#_Toc124962618)

[3.1.1 Complete Scenario of E-commerce trend by comparing sales over month-on-month format 9](#_Toc124962619)

[3.1.2 Complete Scenario of E-commerce trend by comparing sales over month-year format 10](#_Toc124962620)

[3.1.3 Complete Scenario of E-commerce trend by comparing sales over only month format and calculating the percentage difference by comparing with previous month 12](#_Toc124962621)

[3.2 What time do Brazilian customers tend to buy? 14](#_Toc124962622)

[CHAPTER 4: EVOLUTION OF E-COMMERCE IN BRAZIL REGION 16](#_Toc124962623)

[4.1 E-commerce trend by comparison between states and time 16](#_Toc124962624)

[4.1.1 Get month on month orders by states 16](#_Toc124962625)

[4.2 Distribution of customers across the states in Brazil: 19](#_Toc124962626)

[CHAPTER 5: IMPACT ON ECONOMY: ANALYZE THE MONEY MOVEMENT BY E-COMMERCE BY LOOKING AT ORDER PRICES, FREIGHT AND OTHERS 21](#_Toc124962627)

[5.1 % increase in cost of orders 21](#_Toc124962628)

[5.1.1 Get % increase in cost of orders from 2017 to 2018 month over month (include months between Jan to Aug only) - You can use “payment\_value” column in payments table……………………………………………………………………………………..21](#_Toc124962629)

[5.1.2 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 23](#_Toc124962630)

[5.2 mean, sum OF price AND freight value BY customer states 25](#_Toc124962631)

[CHAPTER 6: ANALYSIS ON SALES, FREIGHT AND DELIVERY TIME 27](#_Toc124962632)

[6.1 Calculate days between purchasing, delivering and estimated delivery 27](#_Toc124962633)

[6.2 Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery 29](#_Toc124962634)

[6.3 Sort the data to get the following: 31](#_Toc124962635)

[6.3.1 Top 5 states with lowest average freight value - sort in ASC limit 5 31](#_Toc124962636)

[6.3.2 Top 5 states with highest average freight value - sort in DESC limit 5 32](#_Toc124962637)

[6.3.3 Top 5 states with highest average time to delivery- sort in desc limit 5 33](#_Toc124962638)

[6.3.4 Top 5 states with lowest average time to delivery - sort in asc limit 5 34](#_Toc124962639)

[6.3.5 Top 5 states where delivery is really not so fast compared to estimated date - sort in desc limit 5 34](#_Toc124962640)

[6.3.6 Top 5 states where delivery is really fast compared to estimated date - sort in asc limit 5 35](#_Toc124962641)

[CHAPTER 7: PAYMENT TYPE ANALYSIS 37](#_Toc124962642)

[7.1 Month over Month count of orders for different payment types 37](#_Toc124962643)

[7.1.1 Month over Month count of orders for different payment types 37](#_Toc124962644)

[7.1.2 Month-year vs count OF orders FOR different payment types 39](#_Toc124962645)

[**7.2** Count OF orders based ON the no. OF payment installments ……………………………………………………………………………………...40](#_Toc124962646)

[CHAPTER 8: ACTIONABLE INSIGHTS 43](#_Toc124962647)

[CHAPTER 9: RECOMMENDATIONS 44](#_Toc124962648)

# 

# INTRODUCTION

In Scaler Data Science, Artificial Intelligence and Machine Learning course, first homework project assigned on TARGET store dataset. It is preferred to be solved by using Google’s BigQuery Sandbox SQL platform. Last date to submit the project is 18th January, 2023.

## INTRODUCTION TO TARGET CORPORATION

Target Corporation is a multi-national company headquartered in Minneapolis, Minnesota, USA. It is founded by George Dayton, Douglas Dayton & John Geisse. Previously Target Corporation is known as Dayton Corporation which was founded on June 24, 1902. It is a general merchandise retailer with 1,948 stores in US, 51 supply chain facilities in the US, 1 global capabilities centre in Bengaluru, India and nearly 20 sourcing offices globally.

Target Tag line is “Expect More, Pay Less”. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

## business case details

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. Data is available in 8 Tables in .csv (Comma delimited) format. Tables are named as following:

1. customers.csv
2. geolocation.csv
3. orders.csv
4. order\_items.csv
5. order.reviews.csv
6. payments.csv
7. products.csv
8. sellers.csv

### Features of customers.csv

|  |  |
| --- | --- |
| Features | Description |
| customer\_id | Id of the consumer who made the purchase. |
| customer\_unique\_id | Unique Id of the consumer. |
| customer\_zip\_code\_prefix | Zip Code of the location of the consumer. |
| customer\_city | Name of the City from where order is made. |
| customer\_state | State Code from where order is made (Ex- Sao Paulo-SP) |

### Features of geolocations.csv

|  |  |
| --- | --- |
| Features | Description |
| geolocation\_zip\_code\_prefix | First 5 digits of zip code |
| geolocation\_lat | Latitude |
| geolocation\_lng | Longitude |
| geolocation\_city | City name |
| geolocation\_state | State |

### Features of orders.csv

|  |  |
| --- | --- |
| Features | Description |
| order\_id | A unique id of order made by the consumers. |
| customer\_id | Id of the consumer who made the purchase. |
| order\_status | Status of the order made i.e delivered, shipped etc. |
| order\_purchase\_timestamp | Timestamp of the purchase. |
| order\_delivered\_carrier\_date | Delivery date at which carrier made the delivery. |
| order\_delivered\_customer\_date | Date at which customer got the product. |
| order\_estimated\_delivery\_date | Estimated delivery date of the products. |

### Features of order\_items.csv

|  |  |
| --- | --- |
| Features | Description |
| order\_id | A unique id of order made by the consumers. |
| order\_item\_id | A unique id given to each item ordered in the order. |
| product\_id | A unique id given to each product available on the site. |
| seller\_id | A unique Id of the seller registered in Target. |
| shipping\_limit\_date | The date before which shipping of the ordered product must be completed. |
| price | Actual price of the products ordered. |
| freight\_value | Price rate at which a product is delivered from one point to another. |

### Features of order\_reviews.csv

|  |  |
| --- | --- |
| Features | Description |
| review\_id | Id of the review given on the product ordered by the order id. |
| order\_id | A unique id of order made by the consumers. |
| review\_score | Review score given by the customer for each order on the scale of 1–5. |
| review\_comment\_title | Title of the review |
| review\_comment\_message | Review comments posted by the consumer for each order. |
| review\_creation\_date | Timestamp of the review when it is created. |
| review\_answer\_timestamp | Timestamp of the review answered. |

### Features of payments.csv

|  |  |
| --- | --- |
| Features | Description |
| order\_id | A unique id of order made by the consumers. |
| payment\_sequential | Sequences of the payments made in case of EMI. |
| payment\_type | Mode of payment used. (Ex-Credit Card) |
| payment\_installments | Number of installments in case of EMI purchase. |
| payment\_value | Total amount paid for the purchase order. |

### Features of products.csv

|  |  |
| --- | --- |
| Features | Description |
| product\_id | A unique identifier for the proposed project. |
| product\_category\_name | Name of the product category |
| product\_name\_length | length of the string which specifies the name given to the products ordered. |
| product\_description\_length | length of the description written for each product ordered on the site. |
| product\_photos\_qty | Number of photos of each product ordered available on the shopping portal. |
| product\_weight\_g | Weight of the products ordered in grams. |
| product\_length\_cm | Length of the products ordered in centimetres. |
| product\_height\_cm | Height of the products ordered in centimetres. |
| product\_width\_cm | width of the product ordered in centimetres. |

### Features of sellers.csv

|  |  |
| --- | --- |
| Features | Description |
| seller\_id | Unique Id of the seller registered |
| seller\_zip\_code\_prefix | Zip Code of the location of the seller. |
| seller\_city | Name of the City of the seller. |
| seller\_state | State Code (Ex- Sao Paulo-SP) |

## Entity Relationship Diagram

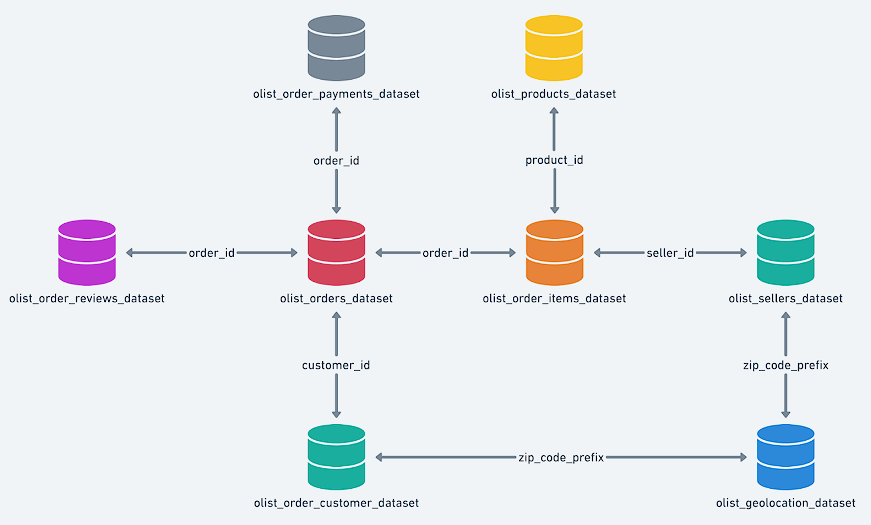


Figure 1‑1 High level overview of relationship between datasets

# IMPORTING THE DATASET AND PRELIMINARY EXPLORATORY ANALYSIS

## Importing the dataset into google’s bigquery sandbox

Firstly, Brazil’s Target Store Data set is provided by Scaler in 8 Tables in .csv format through the [Google Drive link](https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb). Download them to local storage. After signing in Google’s BigQuery Sandbox SQL platform using a Gmail account, create a new project named as “scaler-ds-ai-ml-projects”. Create a new data set/schema under “scaler-ds-ai-ml-projects” named as “TARGET\_BRAZIL\_DATA”. One after another, create 8 tables under “target\_brazil\_data” schema by uploading 8 .csv format files (comma delimited) provided by Scaler. Use auto-detect schema option while creating tables.

### Why BigQuery Sandbox?

BigQuery Sandbox is a SQL platform having following characteristics –

1. Can be easily accessed through internet from anywhere using a Gmail account and using the [BigQuery Sandbox link](https://console.cloud.google.com/bigquery).
2. Free to use without providing any credit card information
3. Can load Big Data in short amount of time (MySQL does not support Big Data, takes lot of time while importing the Big Data)
4. Sandbox gives 10GB of active storage and 1TB of processed query data per month
5. All datasets have default table expiration time. (About 60 days)
6. Free license does not support streaming data, data manipulation language and data transfer services.

## Datatypes of Every column in all tables

**Query:**

SELECT

  TABLE\_CATALOG,

  TABLE\_SCHEMA,

  TABLE\_NAME,

  COLUMN\_NAME,

  IS\_NULLABLE,

  DATA\_TYPE

FROM

  TARGET\_BRAZIL\_DATA.INFORMATION\_SCHEMA.COLUMNS

**Result:**

Table 2.1 Information\_schema.columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TABLE\_CATALOG | TABLE\_SCHEMA | TABLE\_NAME | COLUMN\_NAME | IS\_NULLABLE | DATA\_TYPE |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_REVIEWS | review\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_REVIEWS | order\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_REVIEWS | review\_score | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_REVIEWS | review\_comment\_title | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_REVIEWS | review\_creation\_date | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_REVIEWS | review\_answer\_timestamp | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_category | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_name\_length | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_description\_length | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_photos\_qty | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_weight\_g | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_length\_cm | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_height\_cm | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PRODUCTS | product\_width\_cm | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | SELLERS | seller\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | SELLERS | seller\_zip\_code\_prefix | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | SELLERS | seller\_city | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | SELLERS | seller\_state | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | GEOLOCATION | geolocation\_zip\_code\_prefix | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | GEOLOCATION | geolocation\_lat | YES | FLOAT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | GEOLOCATION | geolocation\_lng | YES | FLOAT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | GEOLOCATION | geolocation\_city | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | GEOLOCATION | geolocation\_state | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | CUSTOMERS | customer\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | CUSTOMERS | customer\_unique\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | CUSTOMERS | customer\_zip\_code\_prefix | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | CUSTOMERS | customer\_city | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | CUSTOMERS | customer\_state | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PAYMENTS | order\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PAYMENTS | payment\_sequential | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PAYMENTS | payment\_type | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PAYMENTS | payment\_installments | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | PAYMENTS | payment\_value | YES | FLOAT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | customer\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_status | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_purchase\_timestamp | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_approved\_at | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_delivered\_carrier\_date | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_delivered\_customer\_date | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDERS | order\_estimated\_delivery\_date | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | order\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | order\_item\_id | YES | INT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | product\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | seller\_id | YES | STRING |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | shipping\_limit\_date | YES | TIMESTAMP |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | price | YES | FLOAT64 |
| scaler-ds-ai-ml-projects | TARGET\_BRAZIL\_DATA | ORDER\_ITEMS | freight\_value | YES | FLOAT64 |

**Insights or Observations:**

1. Only 4 variants of data types are used.
2. All date related attributes are in TIMESTAMP format (YYYY:MM:DD HH:MM:SS.SSSSSS UTC) (UTC – Universal Time Coordinated or Greenwich Meridian Time).
3. All attributes having numbers without decimals are INT64 format.
4. All attributes having numbers with decimals are FLOAT64 format.
5. All other attributes are in STRING format.

## Finding primary keys of each and every table

**Query:**

# Number of customer id's in customers table

SELECT

  COUNT(c.customer\_id)

FROM

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` as c;

# Number of Distinct customer\_id's in customer table

SELECT

  COUNT(DISTINCT c.customer\_id)

FROM

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` as c;

# How many null values are present in customer\_id column

SELECT

  COUNT(c.customer\_id)

FROM

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

WHERE

  c.customer\_id IS NULL;

**Result:**

Number of customer\_id’s in customers table = 99441

Number of Distinct customer\_id’s in customer table = 99441

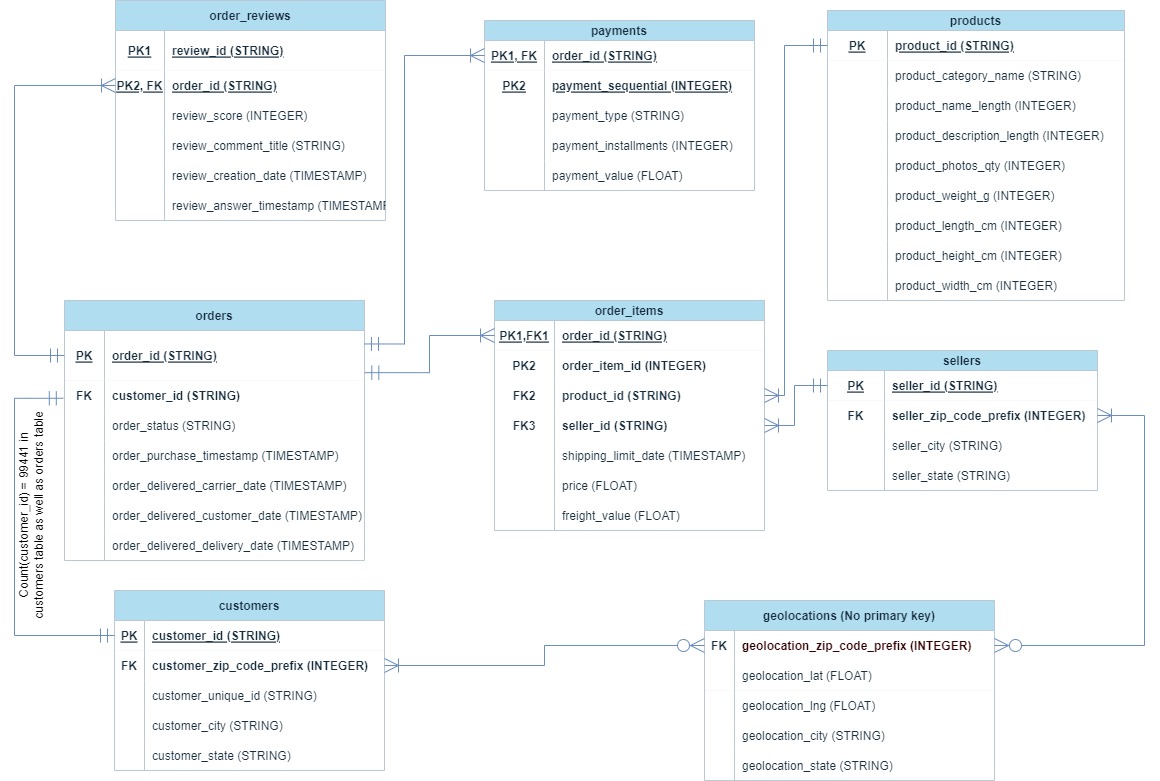
Number of Null values in customer\_id columns = 0

**Insights or Observations:**

1. As number of customer\_id’s and number of distinct customer\_id’s are equal (unique) and there are no null values in the column, customer\_id act as primary key for customers table.
2. Similarly, same procedure can be applied on each and every table to find primary keys.
3. Geolocation table is an exception for which there is no primary key. A surrogate key should be generated in geolocation table.

Table 2.2 Primary keys of each and every table

|  |  |
| --- | --- |
| Tables | Primary Keys |
| customers | customer\_id |
| geolocations | No primary key |
| orders | order\_id |
| order\_items | CONCAT(order\_id, order\_item\_id) |
| order\_reviews | CONCAT(review\_id, order\_id) |
| payments | CONCAT(order\_id, payment\_sequential) |
| products | product\_id |
| sellers | seller id |



## Time period of given data

**Query:**

SELECT

  MIN(order\_purchase\_timestamp) AS Date\_of\_first\_order\_purchased,

  MAX(order\_purchase\_timestamp) AS Date\_of\_last\_order\_purchased,

  MAX(order\_delivered\_customer\_date) AS Date\_of\_last\_order\_delivered,

  DATE\_DIFF(MAX(order\_purchase\_timestamp),MIN(order\_purchase\_timestamp),DAY) AS Time\_period\_of\_given\_data

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS`

**Result :**

|  |  |  |  |
| --- | --- | --- | --- |
| Date\_of\_first\_order\_purchase | Date\_of\_last\_order\_purchase | Date\_of\_last\_order\_delivered | Time\_period\_of\_given\_data |
| 2016-09-04 21:15:19.000000 UTC | 2018-10-17 17:30:18.000000 UTC | 2018-10-17 13:22:46.000000 UTC | 772 |

**Insights or Observations:**

1. Date of first order purchased was 4th September, 2016 and Date of last order purchased was 17th October, 2018.
2. Date of last order purchased and date of last order delivered were same (17th October, 2018).
3. Time period of the given data is 772 days i.e., 2 Years 42 days.

## Distinct Cities and States of Customers who ordered in 772 days’ time period

**Query:**

  # Cities and States of customers ordered during the given period

SELECT

  DISTINCT c.customer\_city,

  c.customer\_state

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  o.customer\_id = c.customer\_id;

  # Number of distinct customer cities

SELECT

  COUNT(DISTINCT c.customer\_city) AS COUNT\_OF\_DIFFERENT\_CITIES

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  o.customer\_id = c.customer\_id;

  # Number of distinct customer states

SELECT

  COUNT(DISTINCT c.customer\_state) AS COUNT\_OF\_DIFFERENT\_STATES

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  o.customer\_id = c.customer\_id;

**Result: (first 10 rows only)**

|  |  |
| --- | --- |
| customer\_city | customer\_state |
| rio de janeiro | RJ |
| sao leopoldo | RS |
| general salgado | SP |
| brasilia | DF |
| paranavai | PR |
| cuiaba | MT |
| sao luis | MA |
| maceio | AL |
| hortolandia | SP |
| varzea grande | MT |

1. Number of distinct customer cities = 4119
2. Number of distinct customer states = 27

**Insights or Observations:**

1. Customers from 4119 cities and 27 states are ordered in Brazil Target stores from 2016 to 2018

# IN-DEPTH EXPLORATION

## Complete Scenario of E-commerce trend (sales over TIME)

### Complete Scenario of E-commerce trend by comparing sales over month-on-month format

 # Create a temporary table using WITH, which changes purchase timestamp into Year & month columns for **SUCCESSFULLY DELIVERED** order\_ids from orders table.

WITH month\_order\_id AS

(

SELECT

  o.order\_id as order\_id,

  EXTRACT(MONTH FROM o.order\_purchase\_timestamp) as month,

  EXTRACT(YEAR FROM o.order\_purchase\_timestamp) as year

FROM `TARGET\_BRAZIL\_DATA.ORDERS` as o

WHERE o.order\_status = "delivered"

ORDER BY month

),

# Create a temporary table using WITH which finds sales by grouping order\_id in payments table

Group\_order\_id\_sales AS

(

  SELECT

    p.order\_id,

    SUM(p.payment\_value) as sales\_by\_order\_id

  FROM `TARGET\_BRAZIL\_DATA.PAYMENTS` as p

  GROUP BY p.order\_id

)

# Query for month on month vs sales

SELECT

  myo.month,

  myo.year,

  SUM(gois.sales\_by\_order\_id) AS monthly\_sales

FROM

  month\_order\_id as myo

LEFT JOIN

  Group\_order\_id\_sales as gois ON gois.order\_id = myo.order\_id

GROUP BY

  myo.month,myo.year

HAVING monthly\_sales IS NOT NULL

ORDER BY myo.month,myo.year ASC;

**Result:**

Table 3.1 E-commerce trend by comparing sales over month-on-month format

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| month | year | monthly\_sales | month | year | monthly\_sales |
| 1 | 2017 | 127545.67 | **7** | 2017 | 566403.93 |
| 1 | 2018 | 1078606.86 | **7** | 2018 | 1027903.86 |
| 2 | 2017 | 271298.65 | **8** | 2017 | 646000.61 |
| 2 | 2018 | 966510.88 | **8** | 2018 | 985414.28 |
| 3 | 2017 | 414369.39 | **9** | 2017 | 701169.99 |
| 3 | 2018 | 1120678 | **10** | 2016 | 46566.71 |
| 4 | 2017 | 390952.18 | **10** | 2017 | 751140.27 |
| 4 | 2018 | 1132933.95 | **11** | 2017 | 1153528.05 |
| 5 | 2017 | 567066.73 | **12** | 2016 | 19.62 |
| 5 | 2018 | 1128836.69 | **12** | 2017 | 843199.17 |
| 6 | 2017 | 490225.6 |
| 6 | 2018 | 1012090.68 |

Figure 3‑1 E-commerce trend by comparing sales over month-on-month format

**Insights or Observations:**

1. Significant number of successful sales were happened from January 2017 to August 2018. Out of this window of time, very negligible sales were happened.
2. In 2017, Sales was gradually increased from January to December. November 2017 is the month where the highest sales are occurred.
3. In 2018, Sales was maintained approximately similar to average value which is around 1000000 dollars.

### Complete Scenario of E-commerce trend by comparing sales over month-year format

**Query:**

 # Create a temporary table using WITH which changes purchase timestamp into Year-month format for every order\_id which is delivered successfully from orders table.

WITH

  month\_year\_order\_id AS (

  SELECT

    o.order\_id AS order\_id,

    FORMAT\_TIMESTAMP('%Y-%m',o.order\_purchase\_timestamp) AS month\_year

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  WHERE

    o.order\_status = "delivered"

  ORDER BY

    month\_year ),

# Create a temporary table using WITH, which finds sales by grouping order\_id in

payments table

  Group\_order\_id\_sales AS (

  SELECT

    p.order\_id,

    SUM(p.payment\_value) AS sales\_by\_order\_id

  FROM

    `TARGET\_BRAZIL\_DATA.PAYMENTS` AS p

  GROUP BY

    p.order\_id )

# Query for month\_year and monthly sales group by and order by month-year

SELECT

  myo.month\_year,

  SUM(gois.sales\_by\_order\_id) AS monthly\_sales

FROM

  month\_year\_order\_id AS myo

LEFT JOIN

  Group\_order\_id\_sales AS gois

ON

  gois.order\_id = myo.order\_id

GROUP BY

  myo.month\_year

HAVING

  monthly\_sales IS NOT NULL

ORDER BY

  myo.month\_year ASC;

**Result:**

Table 3.2 Monthly sales vs Month-year (considering only successfully delivered orders)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | month\_year | monthly\_sales | | 2016-10 | **46566.71** | | 2016-12 | **19.62** | | 2017-01 | **127545.67** | | 2017-02 | **271298.65** | | 2017-03 | **414369.39** | | 2017-04 | **390952.18** | | 2017-05 | **567066.73** | | 2017-06 | **490225.6** | | 2017-07 | **566403.93** | | 2017-08 | **646000.61** | | 2017-09 | **701169.99** | | |  |  | | --- | --- | | month\_year | monthly\_sales | | 2017-10 | **751140.27** | | 2017-11 | **1153528.05** | | 2017-12 | **843199.17** | | 2018-01 | **1078606.86** | | 2018-02 | **966510.88** | | 2018-03 | **1120678** | | 2018-04 | **1132933.95** | | 2018-05 | **1128836.69** | | 2018-06 | **1012090.68** | | 2018-07 | **1027903.86** | | 2018-08 | **985414.28** | |

Figure 3‑2 E-commerce trend by comparing sales over month-year format

**Insights or Observations:**

1. 2016 sales were interestingly very negligible compared to other. It represents something effected hugely on sales like prolonged recession or inflation or bankruptcy or political-military reforms
2. But 2017 was a year where sales were picked up gradually growth.

### Complete Scenario of E-commerce trend by comparing sales over only month format and calculating the percentage difference by comparing with previous month

**Query:**

 # Create a temporary table using WITH which changes purchase timestamp into Only month format for every order\_id which is delivered successfully from orders table.

WITH

  month\_order\_id AS (

  SELECT

    o.order\_id AS order\_id,

    EXTRACT(MONTH

    FROM

      o.order\_purchase\_timestamp) AS month

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  WHERE

    o.order\_status = "delivered"

  ORDER BY

    month ),

# Create a temporary table using WITH, which finds sales by grouping order\_id in

payments table

  Group\_order\_id\_sales AS (

  SELECT

    p.order\_id,

    SUM(p.payment\_value) AS sales\_by\_order\_id

  FROM

    `TARGET\_BRAZIL\_DATA.PAYMENTS` AS p

  GROUP BY

    p.order\_id )

# Query for month and monthly sales group by and order by month

SELECT

  myo.month,

  SUM(gois.sales\_by\_order\_id) AS monthly\_sales

FROM

  month\_order\_id AS myo

LEFT JOIN

  Group\_order\_id\_sales AS gois

ON

  gois.order\_id = myo.order\_id

GROUP BY

  myo.month

HAVING

  monthly\_sales IS NOT NULL

ORDER BY

  myo.month ASC;

**Result:**

Table 3.2 E-commerce trend by comparing sales over only month format and calculating the percentage difference by comparing with previous month

|  |  |  |
| --- | --- | --- |
| month | monthly\_sales | percent difference (calculated using excel) |
| 1 | **1206152.53** | 0 |
| 2 | **1237809.53** | 2.624626588 |
| 3 | **1535047.39** | 24.0132147 |
| 4 | **1523886.13** | -0.727095468 |
| 5 | **1695903.42** | 11.28806717 |
| 6 | **1502316.28** | -11.41498612 |
| 7 | **1594307.79** | 6.123311797 |
| 8 | **1631414.89** | 2.327474044 |
| 9 | **701169.99** | -57.02074351 |
| 10 | **797706.98** | 13.76798656 |
| 11 | **1153528.05** | 44.60548534 |
| 12 | **843218.79** | -26.90088551 |
| Total sales | **15422461.77** |  |

Figure 3‑3 E-commerce trend by comparing sales over only month format and calculating the percentage difference by comparing with previous month

**Insights or observations:**

1. Among all months, September month is the month with lowest sales happened in all three years.
2. May month is the best month for sales cumulatively may be because of some festivals like CARNIVAL, Parintins Folklore fest, Festa Junaina etc.,

## What time do Brazilian customers tend to buy?

**Query:**

   /\* # Brasilia time = UTC - 3:00

# Dawn : 02:00:00 TO 06:59:59

# Morning : 07:00:00 TO 11:59:59

# Afternoon : 12:00:00 TO 16:59:59

# Evening : 17:00:00 TO 20:59:59

# Night : 21:00:00 TO 01:59:59

# Total day divided into 5 equal parts (approx.) \*/

SELECT

  part\_of\_the\_day,

  COUNT(\*) AS No\_of\_orders

FROM (

  SELECT

    purchase\_time,

    CASE

      WHEN purchase\_time BETWEEN "02:00:00" AND "06:59:59" THEN "Dawn"

      WHEN purchase\_time BETWEEN "07:00:00" AND "11:59:59" THEN "Morning"

      WHEN purchase\_time BETWEEN "12:00:00" AND "16:59:59" THEN "Afternoon"

      WHEN purchase\_time BETWEEN "17:00:00" AND "20:59:59" THEN "Evening"

    ELSE

    "Night"

  END

    AS part\_of\_the\_day

  FROM (

    SELECT

      EXTRACT(TIME

      FROM (TIMESTAMP\_SUB(o.order\_purchase\_timestamp,INTERVAL 180 MINUTE))) AS purchase\_time

    FROM

      `TARGET\_BRAZIL\_DATA.ORDERS` AS o) AS Brasilia\_purchase\_time)

GROUP BY

  part\_of\_the\_day;

**# WITHOUT CONVERTING THE TIMEZONES**

SELECT

  part\_of\_the\_day,

  COUNT(\*) AS No\_of\_orders

FROM (

  SELECT

    purchase\_time,

    CASE

      WHEN purchase\_time BETWEEN "02:00:00" AND "06:59:59" THEN "Dawn"

      WHEN purchase\_time BETWEEN "07:00:00" AND "11:59:59" THEN "Morning"

      WHEN purchase\_time BETWEEN "12:00:00" AND "16:59:59" THEN "Afternoon"

      WHEN purchase\_time BETWEEN "17:00:00" AND "20:59:59" THEN "Evening"

    ELSE

    "Night"

  END

    AS part\_of\_the\_day

  FROM (

    SELECT

      EXTRACT(TIME

      FROM (o.order\_purchase\_timestamp)) AS purchase\_time

    FROM

      `TARGET\_BRAZIL\_DATA.ORDERS` AS o) AS Brasilia\_purchase\_time)

GROUP BY

  part\_of\_the\_day;

**Result:**

Table 3.3 Best Phase of the day for sales (changing the UTC timezone to Brasilia timezone )

|  |  |
| --- | --- |
| part\_of\_the\_day | No\_of\_orders |
| Dawn | 9673 |
| Morning | 31837 |
| Afternoon | 31030 |
| Evening | 22349 |
| Night | 4552 |

Table 3.4 Best phase of the day for sales (Without changing the timezones )

|  |  |
| --- | --- |
| part\_of\_the\_day | No\_of\_orders |
| Dawn | 1678 |
| Morning | 21738 |
| Afternoon | 32211 |
| Evening | 24094 |
| Night | 19720 |

**Insights or observations:**

1. According Brasilia timezone , Morning hours from 7 AM to 11:59 AM was the best phase of the day where higher sales recorded.

2. According UTC timezone, Afternoon hours from 12 PM to 04:59 PM was the best phase of the day where higher sales recorded.

3. Dawn and Night hours were not good time for sales (It represents that some effective offers should be given to customers in these hours to increase the sales.)

# EVOLUTION OF E-COMMERCE IN BRAZIL REGION

## E-commerce trend by comparison between states and time

### Get month on month orders by states

# month on month orders by states

SELECT

  month,

  year,

  customer\_state,

  COUNT(\*) as No\_of\_orders

FROM (SELECT

  o.order\_id,

  EXTRACT(MONTH from o.order\_purchase\_timestamp) as month,

  EXTRACT(YEAR from o.order\_purchase\_timestamp) as year,

  c.customer\_state

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` as o

LEFT JOIN `TARGET\_BRAZIL\_DATA.CUSTOMERS` as c ON c.customer\_id = o.customer\_id)

GROUP BY month,year,customer\_state

ORDER BY month,year,customer\_state;

**Result: (in pivoted format)**

Table 4.1 month on month orders by states in pivoted format using excel

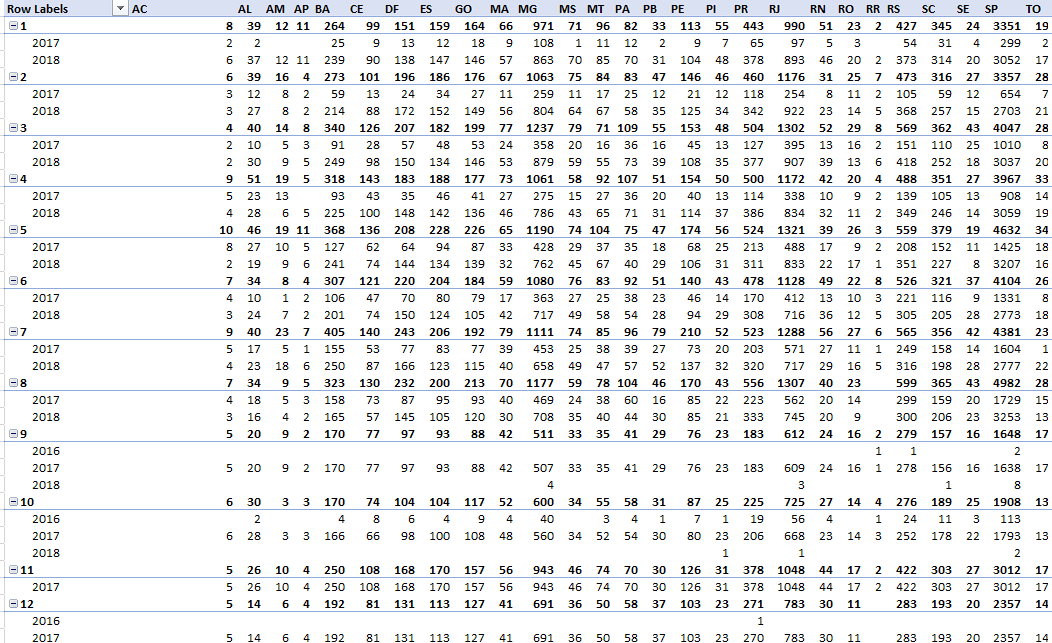


Figure 4‑1 month on month orders by states - 3D Area chart

**Insights or Observations:**

1. State SP 🡪 Sao Paulo dominates other state in sales at every point of time
2. State RJ 🡪 Rio de janeiro was in second position and State MG 🡪 Minas Gerais was in third position.
3. Sharp decrease was observed from august to September at every state.

**Query**

SELECT

  month,

  customer\_state,

  COUNT(\*) AS No\_of\_orders

FROM (

  SELECT

    o.order\_id,

    EXTRACT(MONTH

    FROM

      o.order\_purchase\_timestamp) AS month,

    c.customer\_state

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  LEFT JOIN

    `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

  ON

    c.customer\_id = o.customer\_id)

GROUP BY

  month,

  customer\_state

ORDER BY

  customer\_state,

  month;

**Result:**

Table 4.2 monthly orders by states - pivoted chart using excel

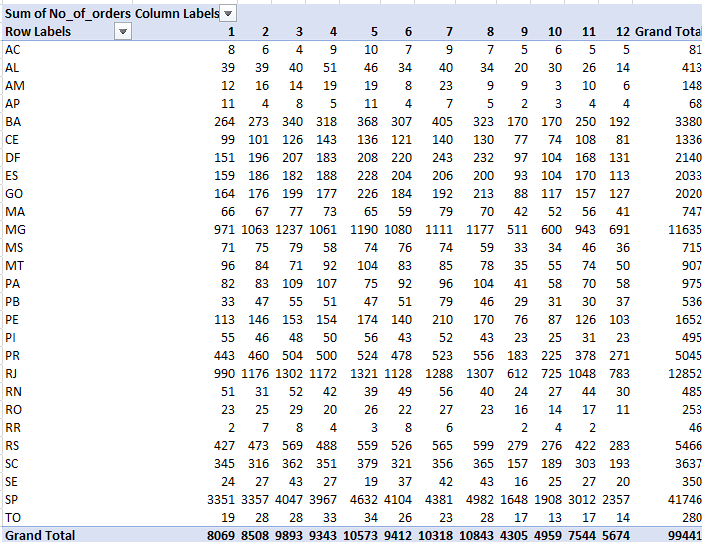
****

Figure 4‑2 month on month orders by states - 100% Stacked area chart

**Insights and observations:**

1. 100% stacked area chart represents relative percentage differences. Combination of all area = 100 % orders = 99441 orders
2. Month 9 is lowest occupied area in stacked chart.
3. Month 8 is highest occupied area in stacked chart.

## Distribution of customers across the states in Brazil:

**Query:**

  # Distribution OF customers across the states

SELECT

  c.customer\_state,

  COUNT(\*) AS No\_of\_customers

FROM

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

GROUP BY

  c.customer\_state

ORDER BY

  c.customer\_state

**Result:**

Table 4.3 Distribution of customers across the states in Brazil

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| customer\_state | No\_of\_customers | customer\_state | No\_of\_customers | customer\_state | No\_of\_customers |
| AC | **81** | **MA** | **747** | **RJ** | **12852** |
| AL | **413** | **MG** | **11635** | **RN** | **485** |
| AM | **148** | **MS** | **715** | **RO** | **253** |
| AP | **68** | **MT** | **907** | **RR** | **46** |
| BA | **3380** | **PA** | **975** | **RS** | **5466** |
| CE | **1336** | **PB** | **536** | **SC** | **3637** |
| DF | **2140** | **PE** | **1652** | **SE** | **350** |
| ES | **2033** | **PI** | **495** | **SP** | **41746** |
| GO | **2020** | **PR** | **5045** | **TO** | **280** |

Figure 4‑3 Distribution of customers across the states in Brazil

**Insights or observations:**

1. Number of customers are dominatingly high in SP🡪 Sao Paulo State. This results in higher sales in that state.

2. Next best states are MG and RJ, which were also plays major role in getting higher sales.

3. So Number of customers should be increased to get higher sales. To get more customers, card offers and membership offers should be provided.

# IMPACT ON ECONOMY: ANALYZE THE MONEY MOVEMENT BY E-COMMERCE BY LOOKING AT ORDER PRICES, FREIGHT AND OTHERS

## % increase in cost of orders

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

  /\*# % increase OF cost OF orders FROM 2017 TO 2018 month over month considering FROM jan TO aug use payment value\*/

# Sum of payment value group by order\_id in payments table

WITH

  Total\_cost\_per\_order\_id AS (

  SELECT

    pa.order\_id,

    SUM(pa.payment\_value) AS Total\_cost\_per\_order\_id

  FROM

    `TARGET\_BRAZIL\_DATA.PAYMENTS` AS pa

  GROUP BY

    pa.order\_id

  ORDER BY

    pa.order\_id),

# Extract purchase year and purchase month from orders table where months are filtered form Jan to august only

Year\_month\_order\_id AS (

  SELECT

    o.order\_id,

    EXTRACT(YEAR

    FROM

      o.order\_purchase\_timestamp) AS purchase\_year,

    EXTRACT(MONTH

    FROM

      o.order\_purchase\_timestamp) AS purchase\_month

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  WHERE

    EXTRACT(MONTH

    FROM

      o.order\_purchase\_timestamp) BETWEEN 01

    AND 08)

# Query the Year and month wise sales data

SELECT

  Y.purchase\_month,

  Y.purchase\_year,

  SUM(T.Total\_cost\_per\_order\_id) AS Total\_sales\_per\_year

FROM

  Total\_cost\_per\_order\_id AS T

INNER JOIN

  Year\_month\_order\_id AS Y

ON

  T.order\_id = Y.order\_id

WHERE

  Y.purchase\_year = 2017

  OR Y.purchase\_year = 2018

GROUP BY

  Y.purchase\_month,Y.purchase\_year

ORDER BY

  Y.purchase\_month,Y.purchase\_year

**RESULT:**

Table 5.1 % increase in cost of orders from 2017 to 2018 month over month

|  |  |  |
| --- | --- | --- |
| purchase\_month | **purchase\_year** | **Total\_sales\_per\_year** |
| **1** | 2017 | 138488.04 |
| 1 | 2018 | 1115004.18 |
| 2 | 2017 | 291908.01 |
| 2 | 2018 | 992463.34 |
| 3 | 2017 | 449863.6 |
| 3 | 2018 | 1159652.12 |
| 4 | 2017 | 417788.03 |
| 4 | 2018 | 1160785.48 |
| 5 | 2017 | 592918.82 |
| 5 | 2018 | 1153982.15 |
| 6 | 2017 | 511276.38 |
| 6 | 2018 | 1023880.5 |
| 7 | 2017 | 592382.92 |
| 7 | 2018 | 1066540.75 |
| 8 | 2017 | 674396.32 |
| 8 | 2018 | 1022425.32 |

Figure 5‑1 % increase in cost of orders from 2017 to 2018 month over month

### 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:

  /\*# % increase OF cost OF orders FROM 2017 TO 2018 considering FROM jan TO aug use payment value\*/

# Sum of payment value group by order\_id in payments table

WITH

  Total\_cost\_per\_order\_id AS (

  SELECT

    pa.order\_id,

    SUM(pa.payment\_value) AS Total\_cost\_per\_order\_id

  FROM

    `TARGET\_BRAZIL\_DATA.PAYMENTS` AS pa

  GROUP BY

    pa.order\_id

  ORDER BY

    pa.order\_id),

# Extract purchase year from orders table where months are filtered form Jan to august only

Year\_order\_id AS (

  SELECT

    o.order\_id,

    EXTRACT(YEAR

    FROM

      o.order\_purchase\_timestamp) AS purchase\_year

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  WHERE

    EXTRACT(MONTH

    FROM

      o.order\_purchase\_timestamp) BETWEEN 01

    AND 08)

# Query the Year wise sales data

SELECT

  Y.purchase\_year,

  SUM(T.Total\_cost\_per\_order\_id) AS Total\_sales\_per\_year

FROM

  Total\_cost\_per\_order\_id AS T

INNER JOIN

  Year\_order\_id AS Y

ON

  T.order\_id = Y.order\_id

WHERE

  Y.purchase\_year = 2017

  OR Y.purchase\_year = 2018

GROUP BY

  Y.purchase\_year

**Result:**

Table 5.2 % increase in cost of orders from 2017 to 2018

|  |  |  |
| --- | --- | --- |
| purchase\_year | Total\_sales\_per\_year | % increase |
| 2017 | 3669022 | 0 |
| 2018 | 8694734 | 136.9769 |

Figure 5‑2 % increase in cost of orders from 2017 to 2018

**Insights or observations:**

1. From 2017 to 2018 , 1.3 times increase in sales was observed.

## mean, sum OF price AND freight value BY customer states

Query:

SELECT

  c.customer\_state,

  SUM(oi.price) AS sum\_of\_price,

  AVG(oi.price) AS mean\_of\_price,

  SUM(oi.freight\_value) AS sum\_of\_freight\_value,

  AVG(oi.freight\_value) AS mean\_of\_freight\_value

FROM

  `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

LEFT JOIN

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

ON

  oi.order\_id = o.order\_id

LEFT JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  o.customer\_id = c.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  c.customer\_state;

**Result**:

Table 5.3 Sum and mean of price and freight value for all states

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| customer\_state | sum\_of\_price | mean\_of\_price | sum\_of\_freight\_value | mean\_of\_freight\_value |
| AC | 15982.95 | 173.7277174 | 3686.75 | 40.07336957 |
| AL | 80314.81 | 180.8892117 | 15914.59 | 35.84367117 |
| AM | 22356.84 | 135.496 | 5478.89 | 33.20539394 |
| AP | 13474.3 | 164.3207317 | 2788.5 | 34.00609756 |
| BA | 511349.99 | 134.6012082 | 100156.68 | 26.36395894 |
| CE | 227254.71 | 153.7582612 | 48351.59 | 32.71420162 |
| DF | 302603.94 | 125.7705486 | 50625.5 | 21.04135495 |
| ES | 275037.31 | 121.9137012 | 49764.6 | 22.0587766 |
| GO | 294591.95 | 126.2717317 | 53114.98 | 22.76681526 |
| MA | 119648.22 | 145.2041505 | 31523.77 | 38.25700243 |
| MG | 1585308.03 | 120.7485741 | 270853.46 | 20.63016681 |
| MS | 116812.64 | 142.6283761 | 19144.03 | 23.374884 |
| MT | 156453.53 | 148.2971848 | 29715.43 | 28.16628436 |
| PA | 178947.81 | 165.6924167 | 38699.3 | 35.83268519 |
| PB | 115268.08 | 191.4752159 | 25719.73 | 42.72380399 |
| PE | 262788.03 | 145.5083223 | 59449.66 | 32.91786268 |
| PI | 86914.08 | 160.3580812 | 21218.2 | 39.14797048 |
| PR | 683083.76 | 119.0041394 | 117851.68 | 20.53165157 |
| RJ | 1824092.67 | 125.1178181 | 305589.31 | 20.96092393 |
| RN | 83034.98 | 156.9659357 | 18860.1 | 35.65236295 |
| RO | 46140.64 | 165.9735252 | 11417.38 | 41.06971223 |
| RR | 7829.43 | 150.5659615 | 2235.19 | 42.98442308 |
| RS | 750304.02 | 120.3374531 | 135522.74 | 21.73580433 |
| SC | 520553.34 | 124.6535776 | 89660.26 | 21.47036877 |
| SE | 58920.85 | 153.0411688 | 14111.47 | 36.65316883 |
| SP | 5202955.05 | 109.6536292 | 718723.07 | 15.14727539 |
| TO | 49621.74 | 157.5293333 | 11732.68 | 37.24660317 |

Figure 5‑3 Sum and mean of price and freight value for all states

**Insights or observations:**

1. By observing the trends between mean of price and freight value, both of them are proportional to each other. Similar trend was occurred between them
2. Once again, Sao Paolo, Rio Di jenero and Minas Gerais states with excellent behaviour. The techniques used in these 3 states should be applied on other states too. At all these three states, mean line is dipped down compared to others implies that freight value and prices are significantly lesser compared to others.
3. Highest average of price is occurred at State PB 🡪 Paraiba state. So by decreasing the freight values and price values, we can expect increase in sales.

# ANALYSIS ON SALES, FREIGHT AND DELIVERY TIME

## Calculate days between purchasing, delivering and estimated delivery

**Query:**

  # days BETWEEN purchasing,delivery AND estimated delivery

SELECT

  o.order\_id,

  DATE\_DIFF(DATE(o.order\_delivered\_customer\_date),DATE(o.order\_purchase\_timestamp),DAY) AS Actual\_delivery\_days,

  DATE\_DIFF(DATE(o.order\_estimated\_delivery\_date),DATE(o.order\_purchase\_timestamp),DAY) AS Estimated\_delivery\_days,

  DATE\_DIFF(DATE(o.order\_estimated\_delivery\_date),DATE(o.order\_delivered\_customer\_date),DAY) AS Days\_between\_actual\_and\_estimated

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

WHERE

  o.order\_status = "delivered";

Result:

Table 6.1 Days between purchasing date, actual delivery date and estimated delivery date

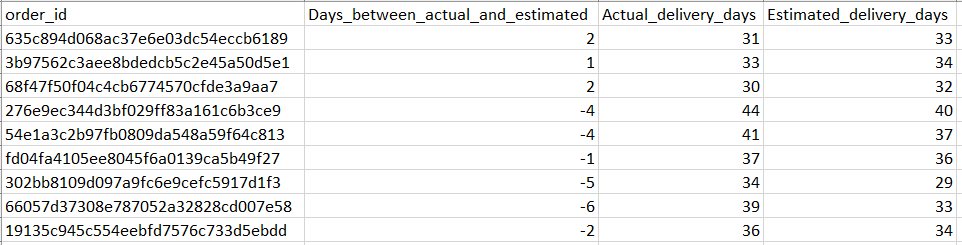


Figure 6‑1 Days between purchasing date, actual delivery date and estimated delivery date

**Insights or observations:**

1. Actual delivery days on average is 12.87 days
2. Estimated delivery days on average is 24.37 days
3. Days between actual and estimated on average is 11.87 days
4. So Estimation algorithm should be updated as it is showing nearly 100% of error.

Query:

  # time BETWEEN purchasing,delivery AND estimated delivery

SELECT

  o.order\_id,

  TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE) AS Actual\_delivery\_in\_minutes,

  TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_purchase\_timestamp),MINUTE) AS Estimated\_delivery\_in\_minutes,

  TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE) AS Minutes\_between\_actual\_and\_estimated

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

WHERE

  o.order\_status = "delivered";

**Result:**

Table 6.2 Time between purchasing, actual delivery and estimated delivery in minutes

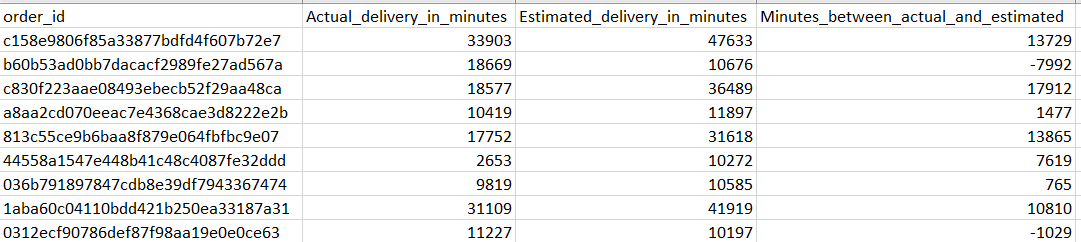


Figure 6‑2 Time between purchasing, actual delivery and estimated delivery in minutes

**Insights and observations:**

1. Sum of actual delivery time and minutes between actual and estimate time = estimate delivery time. Similar behaviour observed in time units also.

## Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

**Query:**

  /\*# GROUP DATA BY state,take mean OF freight\_value,time\_to\_delivery,diff\_estimate d\_delivery\*/

WITH

  total\_freight\_value\_per\_order AS (

  SELECT

    oi.order\_id,

    SUM(oi.freight\_value) AS total\_freight\_value

  FROM

    `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

  GROUP BY

    oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) AS Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) AS time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) AS diff\_estimated\_delivery

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  total\_freight\_value\_per\_order AS tfvpo

ON

  tfvpo.order\_id = o.order\_id

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  c.customer\_id = o.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  c.customer\_state;

**Result:**

Table 6.3 Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimate d\_delivery

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | time\_to\_delivery | diff\_estimated\_delivery | Mean\_of\_freight\_value |
| AC | 30290.9375 | 28910.4875 | 45.5154321 |
| AL | 35342.66751 | 11565.96222 | 38.72163017 |
| AM | 38052.95862 | 27146.22759 | 37.27136054 |
| AP | 39146 | 27445.25373 | 41.00735294 |
| BA | 27842.57463 | 14542.37193 | 29.82628946 |
| CE | 30623.36826 | 14558.13526 | 36.43676714 |
| DF | 18672.80337 | 16328.5625 | 23.82376471 |
| ES | 22736.11278 | 14107.53283 | 24.57511111 |
| GO | 22472.64333 | 16536.00255 | 26.46486298 |
| MA | 31064.59693 | 12798.23989 | 42.59968919 |
| MG | 17294.28789 | 18052.893 | 23.46270444 |
| MS | 22489.92011 | 14911.68474 | 27.00145275 |
| MT | 25999.81264 | 19706.82393 | 32.90745293 |
| PA | 34232.5148 | 19275.94397 | 39.89618557 |
| PB | 29414.03868 | 18142.95358 | 48.34535714 |
| PE | 26565.08977 | 18155.08663 | 36.07382282 |
| PI | 28017.72689 | 15306.26681 | 43.03894523 |
| PR | 17267.3833 | 18171.96019 | 23.57976791 |
| RJ | 22045.98616 | 15926.68032 | 23.94525231 |
| RN | 27760.06329 | 18665.44304 | 39.12883817 |
| RO | 27895.2428 | 27931.04938 | 46.22421053 |
| RR | 42317.5122 | 23895.87805 | 48.59108696 |
| RS | 22031.90232 | 19019.85853 | 24.94895803 |
| SC | 21540.89371 | 15561.30646 | 24.82288483 |
| SE | 30988.01493 | 13432.93433 | 40.90281159 |
| SP | 12615.85949 | 14948.5431 | 17.37095033 |
| TO | 25427.14234 | 16472.91606 | 42.05261649 |

Figure 6‑3 Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimate d\_delivery

**Insights or observations:**

1. It looks like there is proportionaly behaviour between time required to delivery and average freight value.

## Sort the data to get the following:

### Top 5 states with lowest average freight value - sort in ASC limit 5

Query:

  /\*# Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery \*/

WITH

  total\_freight\_value\_per\_order AS (

  SELECT

    oi.order\_id,

    SUM(oi.freight\_value) AS total\_freight\_value

  FROM

    `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

  GROUP BY

    oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) AS Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) AS time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) AS diff\_estimated\_delivery

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  total\_freight\_value\_per\_order AS tfvpo

ON

  tfvpo.order\_id = o.order\_id

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  c.customer\_id = o.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  Mean\_of\_freight\_value ASC

LIMIT

  5

**Result:**

Table 6.4 Top 5 states with lowest average freight value - sort in ASC limit 5

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | Mean\_of\_freight\_value | time\_to\_delivery | diff\_estimated\_delivery |
| SP | 17.37095033 | 12615.85949 | 14948.5431 |
| MG | 23.46270444 | 17294.28789 | 18052.893 |
| PR | 23.57976791 | 17267.3833 | 18171.96019 |
| DF | 23.82376471 | 18672.80337 | 16328.5625 |
| RJ | 23.94525231 | 22045.98616 | 15926.68032 |

**Insights or observations:**

1. Sao Paolo, Minas Gerais, Parana, Distrito Federal and Rio de Janeiro were the bottom 5 states according to mean freight value. That means these are the states where lowest freight value occurred.

### Top 5 states with highest average freight value - sort in DESC limit 5

Query:

WITH

  total\_freight\_value\_per\_order AS (

  SELECT

    oi.order\_id,

    SUM(oi.freight\_value) AS total\_freight\_value

  FROM

    `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

  GROUP BY

    oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) AS Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) AS time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) AS diff\_estimated\_delivery

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  total\_freight\_value\_per\_order AS tfvpo

ON

  tfvpo.order\_id = o.order\_id

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  c.customer\_id = o.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  Mean\_of\_freight\_value DESC

LIMIT

  5

**Result:**

Table 6.5 Top 5 states with highest average freight value - sort in DESC limit 5

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | Mean\_of\_freight\_value | time\_to\_delivery | diff\_estimated\_delivery |
| RR | 48.59108696 | 42317.5122 | 23895.87805 |
| PB | 48.34535714 | 29414.03868 | 18142.95358 |
| RO | 46.22421053 | 27895.2428 | 27931.04938 |
| AC | 45.5154321 | 30290.9375 | 28910.4875 |
| PI | 43.03894523 | 28017.72689 | 15306.26681 |

**Insights or observations:**

1. Roraima, Paraiba, Rondonia, Acre and Piaui were the TOP 5 states where mean freight value was higher.
2. It is needed to decrease the freight value in these states.

### Top 5 states with highest average time to delivery- sort in desc limit 5

Query:

WITH

  total\_freight\_value\_per\_order AS (

  SELECT

    oi.order\_id,

    SUM(oi.freight\_value) AS total\_freight\_value

  FROM

    `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

  GROUP BY

    oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) AS Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) AS time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) AS diff\_estimated\_delivery

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  total\_freight\_value\_per\_order AS tfvpo

ON

  tfvpo.order\_id = o.order\_id

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  c.customer\_id = o.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  time\_to\_delivery DESC

LIMIT

  5

**Result:**

Table 6.6 Top 5 states with highest average time to delivery- sort in desc limit 5

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | Mean\_of\_freight\_value | time\_to\_delivery | diff\_estimated\_delivery |
| RR | 48.59108696 | 42317.5122 | 23895.87805 |
| AP | 41.00735294 | 39146 | 27445.25373 |
| AM | 37.27136054 | 38052.95862 | 27146.22759 |
| AL | 38.72163017 | 35342.66751 | 11565.96222 |
| PA | 39.89618557 | 34232.5148 | 19275.94397 |

**Insights or observations:**

1. Roraima, Amapa, Amazons, Alagoas and Para were the Top 5 states according to the average time required to deliver
2. In these state, It is necessary to increase the speed of delivery

### Top 5 states with lowest average time to delivery - sort in asc limit 5

Query:

WITH total\_freight\_value\_per\_order AS (

SELECT

  oi.order\_id,

  SUM(oi.freight\_value) as total\_freight\_value

FROM

  `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` as oi

GROUP BY oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) as Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) as time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) as diff\_estimated\_delivery

FROM `TARGET\_BRAZIL\_DATA.ORDERS` as o

INNER JOIN total\_freight\_value\_per\_order as tfvpo ON tfvpo.order\_id = o.order\_id

INNER JOIN `TARGET\_BRAZIL\_DATA.CUSTOMERS` as c ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_state

ORDER BY time\_to\_delivery ASC

LIMIT 5

**Result:**

Table 6.7 Top 5 states with lowest average time to delivery - sort in asc limit 5

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | Mean\_of\_freight\_value | time\_to\_delivery | diff\_estimated\_delivery |
| SP | 17.37095033 | 12615.85949 | 14948.5431 |
| PR | 23.57976791 | 17267.3833 | 18171.96019 |
| MG | 23.46270444 | 17294.28789 | 18052.893 |
| DF | 23.82376471 | 18672.80337 | 16328.5625 |
| SC | 24.82288483 | 21540.89371 | 15561.30646 |

**Insights or observations:**

1. Sao Paulo, Parana, Minas Gerais, Distrito Federal and Santa Catarina were the bottom 5 states according to time required for delivery.
2. These are fairly developed according to transportation facilties.

### Top 5 states where delivery is really not so fast compared to estimated date - sort in desc limit 5

Query:

WITH

  total\_freight\_value\_per\_order AS (

  SELECT

    oi.order\_id,

    SUM(oi.freight\_value) AS total\_freight\_value

  FROM

    `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

  GROUP BY

    oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) AS Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) AS time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) AS diff\_estimated\_delivery

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  total\_freight\_value\_per\_order AS tfvpo

ON

  tfvpo.order\_id = o.order\_id

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  c.customer\_id = o.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  diff\_estimated\_delivery DESC

LIMIT

  5

**Result:**

Table 6.8 Top 5 states where delivery is really not so fast compared to estimated date - sort in desc limit 5

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | Mean\_of\_freight\_value | time\_to\_delivery | diff\_estimated\_delivery |
| AC | 45.5154321 | 30290.9375 | 28910.4875 |
| RO | 46.22421053 | 27895.2428 | 27931.04938 |
| AP | 41.00735294 | 39146 | 27445.25373 |
| AM | 37.27136054 | 38052.95862 | 27146.22759 |
| RR | 48.59108696 | 42317.5122 | 23895.87805 |

**Insights or Observations:**

1. Acre, Rondonia, Amapa, Amazons and Roraima are top 5 states where estimated delivery was having high offset.
2. Here Freight value should be controlled by increasing the transportation resources and bring the accuracy in delivery estimation with less freight price.

### Top 5 states where delivery is really fast compared to estimated date - sort in asc limit 5

**Query:**

WITH

  total\_freight\_value\_per\_order AS (

  SELECT

    oi.order\_id,

    SUM(oi.freight\_value) AS total\_freight\_value

  FROM

    `TARGET\_BRAZIL\_DATA.ORDER\_ITEMS` AS oi

  GROUP BY

    oi.order\_id)

SELECT

  c.customer\_state,

  AVG(total\_freight\_value) AS Mean\_of\_freight\_value,

  AVG(TIMESTAMP\_DIFF((o.order\_delivered\_customer\_date),(o.order\_purchase\_timestamp),MINUTE)) AS time\_to\_delivery,

  AVG(TIMESTAMP\_DIFF((o.order\_estimated\_delivery\_date),(o.order\_delivered\_customer\_date),MINUTE)) AS diff\_estimated\_delivery

FROM

  `TARGET\_BRAZIL\_DATA.ORDERS` AS o

INNER JOIN

  total\_freight\_value\_per\_order AS tfvpo

ON

  tfvpo.order\_id = o.order\_id

INNER JOIN

  `TARGET\_BRAZIL\_DATA.CUSTOMERS` AS c

ON

  c.customer\_id = o.customer\_id

GROUP BY

  c.customer\_state

ORDER BY

  diff\_estimated\_delivery ASC

LIMIT

  5

**Result:**

Table 6.9 Top 5 states where delivery is really fast compared to estimated date - sort in asc limit 5

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_state | Mean\_of\_freight\_value | time\_to\_delivery | diff\_estimated\_delivery |
| AL | 38.72163017 | 35342.66751 | 11565.96222 |
| MA | 42.59968919 | 31064.59693 | 12798.23989 |
| SE | 40.90281159 | 30988.01493 | 13432.93433 |
| ES | 24.57511111 | 22736.11278 | 14107.53283 |
| BA | 29.82628946 | 27842.57463 | 14542.37193 |

**Insights or observations:**

1. Alagoas, Maranhao, Serigipe, Espirito santo and Bahia are bottom 5 states where estimated delivery is higly accurate comparatively.
2. But these states are not producing high sales. So sales promotion should be happened here.

# PAYMENT TYPE ANALYSIS

## Month over Month count of orders for different payment types

### Month over Month count of orders for different payment types

WITH

  month\_year\_order\_id AS (

  SELECT

    o.order\_id AS order\_id,

      EXTRACT(YEAR

    FROM

      o.order\_purchase\_timestamp) AS purchase\_year,

    EXTRACT(MONTH

    FROM

      o.order\_purchase\_timestamp) AS purchase\_month

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  ORDER BY

    purchase\_month, purchase\_year )

SELECT

  COUNT(\*) AS Number\_of\_orders,

  myoi.purchase\_month,

  myoi.purchase\_year,

  pa.payment\_type

FROM

  `TARGET\_BRAZIL\_DATA.PAYMENTS` AS pa

INNER JOIN

  month\_year\_order\_id AS myoi

ON

  myoi.order\_id = pa.order\_id

GROUP BY

  pa.payment\_type,

  myoi.purchase\_month,

  myoi.purchase\_year

ORDER BY

  myoi.purchase\_month,

  pa.payment\_type,

  myoi.purchase\_year

Table 7.1 Month over Month count of orders for different payment types (pivot table)

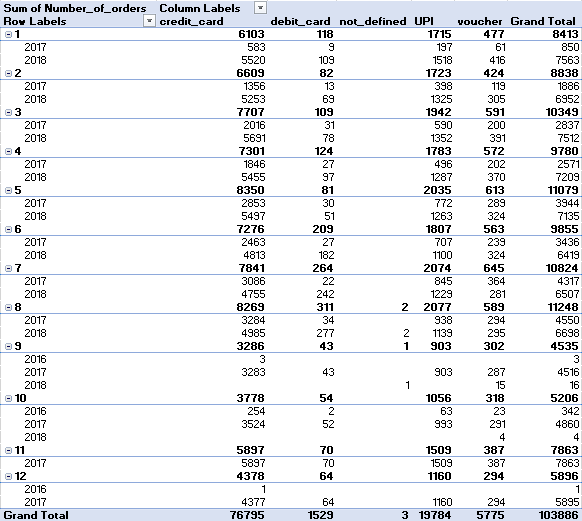


Figure 7‑1 Month over Month count of orders for different payment types (area chart)

**Insights or observations:**

1. More number of payments happened by credit card in every month. Comparitively 2018 had more number of orders.
2. Continuity was missed from month 9 to 12

### Month-year vs count OF orders FOR different payment types

Query:

WITH

  month\_year\_order\_id AS (

  SELECT

    o.order\_id AS order\_id,

    FORMAT\_TIMESTAMP('%Y-%m',o.order\_purchase\_timestamp) AS month\_year

  FROM

    `TARGET\_BRAZIL\_DATA.ORDERS` AS o

  ORDER BY

    month\_year )

SELECT

  COUNT(\*) AS Number\_of\_orders,

  myoi.month\_year,

  pa.payment\_type

FROM

  `TARGET\_BRAZIL\_DATA.PAYMENTS` AS pa

INNER JOIN

  month\_year\_order\_id AS myoi

ON

  myoi.order\_id = pa.order\_id

GROUP BY

  pa.payment\_type,

  myoi.month\_year

ORDER BY

  myoi.month\_year,

  pa.payment\_type

**Result:**

Table 7.2Month-year vs count OF orders FOR different payment types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| month\_year | credit\_card | debit\_card | not\_defined | UPI | voucher |
| 2016-09 | 3 |  |  |  |  |
| 2016-10 | 254 | 2 |  | 63 | 23 |
| 2016-12 | 1 |  |  |  |  |
| 2017-01 | 583 | 9 |  | 197 | 61 |
| 2017-02 | 1356 | 13 |  | 398 | 119 |
| 2017-03 | 2016 | 31 |  | 590 | 200 |
| 2017-04 | 1846 | 27 |  | 496 | 202 |
| 2017-05 | 2853 | 30 |  | 772 | 289 |
| 2017-06 | 2463 | 27 |  | 707 | 239 |
| 2017-07 | 3086 | 22 |  | 845 | 364 |
| 2017-08 | 3284 | 34 |  | 938 | 294 |
| 2017-09 | 3283 | 43 |  | 903 | 287 |
| 2017-10 | 3524 | 52 |  | 993 | 291 |
| 2017-11 | 5897 | 70 |  | 1509 | 387 |
| 2017-12 | 4377 | 64 |  | 1160 | 294 |
| 2018-01 | 5520 | 109 |  | 1518 | 416 |
| 2018-02 | 5253 | 69 |  | 1325 | 305 |
| 2018-03 | 5691 | 78 |  | 1352 | 391 |
| 2018-04 | 5455 | 97 |  | 1287 | 370 |
| 2018-05 | 5497 | 51 |  | 1263 | 324 |
| 2018-06 | 4813 | 182 |  | 1100 | 324 |
| 2018-07 | 4755 | 242 |  | 1229 | 281 |
| 2018-08 | 4985 | 277 | 2 | 1139 | 295 |
| 2018-09 |  |  | 1 |  | 15 |
| 2018-10 |  |  |  |  | 4 |

Figure 7‑2 Month-year vs count OF orders FOR different payment types

**Insights or observations:**

1. More number of payments were happened by using Credit card and then UPI.
2. So to increase the sales rapidly, offeres should be posted for debit card and vouchers
3. Credit card and UPI customers can be attracted for higher cart amount with additional restrictions.
4. Trend on Credit card users and UPI users gradually increased from 2016 to 2018.

## Count OF orders based ON the no. OF payment installments

**Query:**

  # Count OF orders based ON the no. OF payment installments

SELECT

  pa.payment\_installments,

  COUNT(\*) AS Number\_of\_orders

FROM

  `TARGET\_BRAZIL\_DATA.PAYMENTS` AS pa

GROUP BY

  pa.payment\_installments

ORDER BY

  pa.payment\_installments;

**Result:**

Table 7.3 Count OF orders based ON the no. OF payment installments

|  |  |
| --- | --- |
| payment\_installments | Number\_of\_orders |
| 0 | 2 |
| 1 | 52546 |
| 2 | 12413 |
| 3 | 10461 |
| 4 | 7098 |
| 5 | 5239 |
| 6 | 3920 |
| 7 | 1626 |
| 8 | 4268 |
| 9 | 644 |
| 10 | 5328 |
| 11 | 23 |
| 12 | 133 |
| 13 | 16 |
| 14 | 15 |
| 15 | 74 |
| 16 | 5 |
| 17 | 8 |
| 18 | 27 |
| 20 | 17 |
| 21 | 3 |
| 22 | 1 |
| 23 | 1 |
| 24 | 18 |

Figure 7‑3 Count OF orders based ON the no. OF payment installments

**Insights and Observations:**

1. Number of orders were dominatingly more with two installments.
2. Very negligible number of orders were made with installments 1 and greater than 11.
3. It is may be due to high interest beyond 11 months.

# ACTIONABLE INSIGHTS

1. Time period of the given data is 2 Years 42 days only.
2. Total 4119 cities and 27 states customers involved in the sales of Target Store.
3. Last three months of 2016 data looks like the worst period of time for sales of Target store. Huge effected by Prolonged recession or inflation or political reforms.
4. Gradual growth in number of orders and sales observed in 2017.
5. Sales and number of orders growth maintained average behaviour in 2018.
6. Month wise – November 2017 had the highest sales.
7. Cumulatively September month has lowest sales (overall).
8. Best three phases of the day in Descending order are Afternoon 🡪 Morning 🡪 Evening
9. Sao Paolo, Rio De Janeiro and Minas Gerais dominates other states in number of orders and number of customers wise too.
10. From 2017 to 2018, 1.3 times growth was observed.
11. Estimation on delivery days should be improved. As actual and estimated delivery days difference in range of 12 days on average.
12. Sao Paolo was having lowest average freight value and also lowest average delivery time.
13. Roraima was having highest average freight value and also highest average delivery time.
14. Acre was having high offset in estimated delivery.
15. Alagoas was having low offset in estimated delivery.
16. Credit Card payments are in first position and UPI payments are in second postion.
17. More number of customers paid by using 2 installments only.

# RECOMMENDATIONS

1. Sales promotion campaign should be held in states like Roraima, Acre etc.,
2. Voucher offers should be provided to increase the sales through vouchers
3. Less EMI or No cost EMI should be provided to increase the number of installments.
4. Estimation strategy should be modified by using machine learning techniques like time series forecasting using ARIMA such that average offset in estimation should be below 3-4 days
5. Sales are highly happening in April to August time period because of CARNIVAL, Folklore fest etc., So inventory should be optimized well in this period. Big day sales can attract high number of sales.
6. Delivery time directly related to Freight value. So, Infrastructure investment is required in backlog states like Roraima, Acre etc.,
7. Sao Paolo, Rio De Janeiro and Minas Gerais are the three important states where Target should never loose grip on these three states.