**Context**

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

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

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

The data is available in 8 csv files:

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The column description for these csv files is given below.

The **customers.csv** contain following features:

|  |  |
| --- | --- |
| **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 consumer’s location |
| customer\_city | Name of the City from where order is made |
| customer\_state | State Code from where order is made (Eg. são paulo - SP) |

The **sellers.csv** contains following features:

|  |  |
| --- | --- |
| **Features** | **Description** |
| seller\_id | Unique ID of the seller registered |
| seller\_zip\_code\_prefix | Zip Code of the seller’s location |
| seller\_city | Name of the City of the seller |
| seller\_state | State Code (Eg. são paulo - SP) |

The **order\_items.csv** contain following features:

|  |  |
| --- | --- |
| **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 | Unique ID of the seller registered in Target |
| shipping\_limit\_date | The date before which the ordered product must be shipped |
| price | Actual price of the products ordered |
| freight\_value | Price rate at which a product is delivered from one point to another |

The **geolocations.csv** contain following features:

|  |  |
| --- | --- |
| **Features** | **Description** |
| geolocation\_zip\_code\_prefix | First 5 digits of Zip Code |
| geolocation\_lat | Latitude |
| geolocation\_lng | Longitude |
| geolocation\_city | City |
| geolocation\_state | State |

The **payments.csv** contain following features:

|  |  |
| --- | --- |
| **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 (Eg. Credit Card) |
| payment\_installments | Number of installments in case of EMI purchase |
| payment\_value | Total amount paid for the purchase order |

The **orders.csv** contain following features:

|  |  |
| --- | --- |
| **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 |

The **reviews.csv** contain following features:

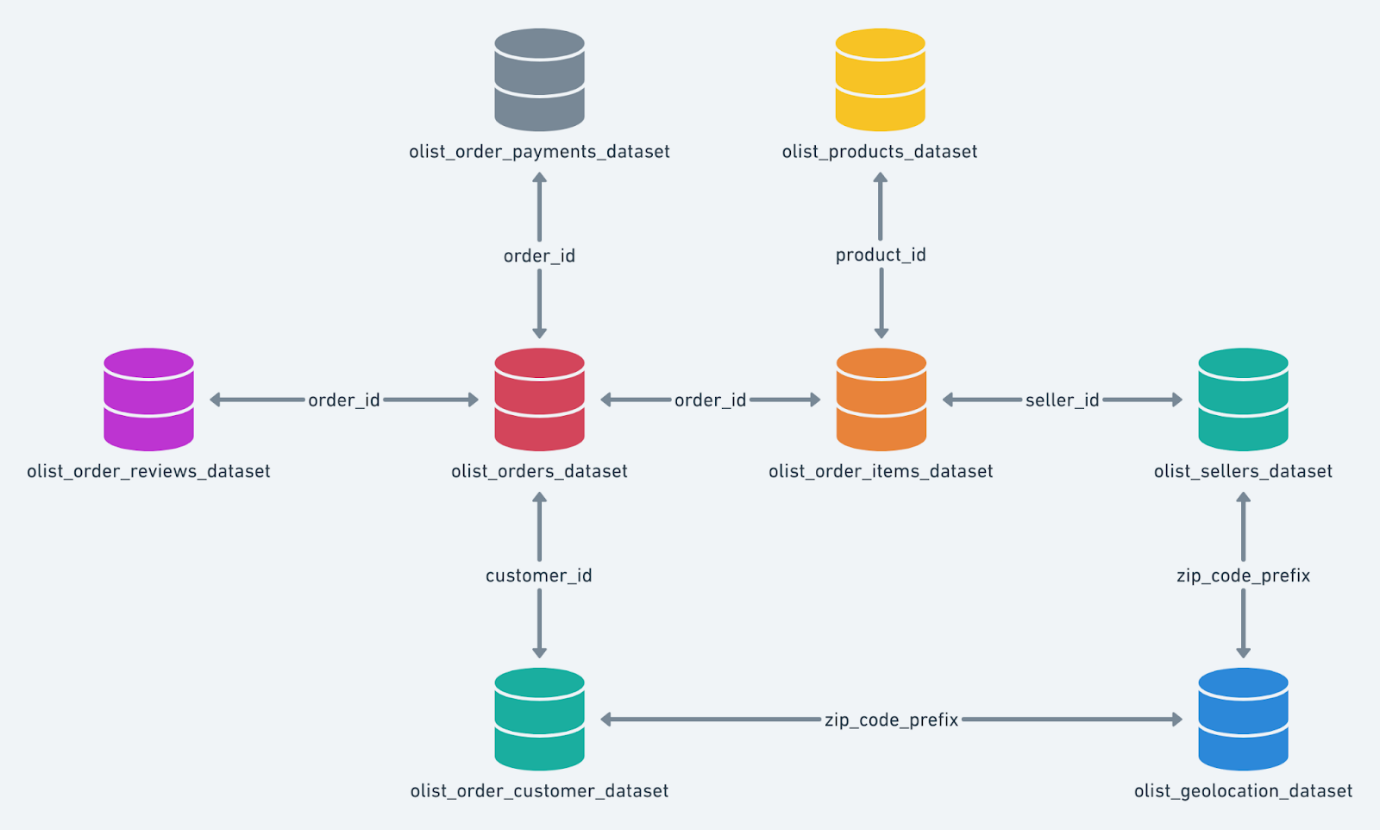
|  |  |
| --- | --- |
| **Features** | **Description** |
| review\_id | ID of the review given on the product ordered by the order id |
| order\_id | A Unique ID of order made by the consumers |
| review\_score | Review score given by the customer for each order on a scale of 1-5 |
| review\_comment\_title | 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 |

The **products.csv** contain following features:

|  |  |
| --- | --- |
| **Features** | **Description** |
| product\_id | A Unique identifier for the proposed project. |
| product\_category\_name | Name of the product category |
| product\_name\_lenght | Length of the string which specifies the name given to the products ordered |
| product\_description\_lenght | 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 centimeters |
| product\_height\_cm | Height of the products ordered in centimeters |
| product\_width\_cm | Width of the product ordered in centimeters |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dataset schema:**



**Question 1**

Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.  
  
1.1 **Data type of columns in a table.**  
  
Run below query to get result set containing table name, column name, column data type and isNullable flag.

SELECT

  table\_name,

  column\_name,

  data\_type,

  is\_nullable

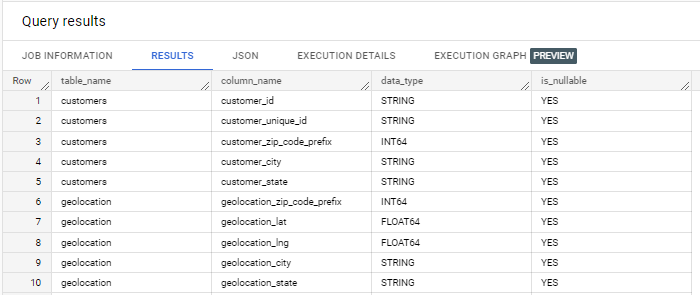
FROM

  `total-scion-382610.target\_sql\_business\_case.I**NFORMATION\_SCHEMA.COLUMNS**`

ORDER BY

  table\_name,

  ordinal\_position



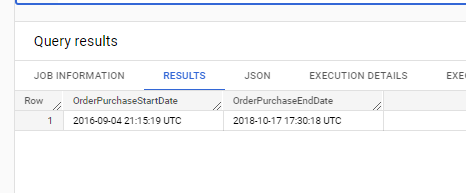
1.2 **Time period for which the data is given.**  
  
SELECT

  min(order\_purchase\_timestamp) AS OrderPurchaseStartDate,

  max(order\_purchase\_timestamp) AS OrderPurchaseEndDate

 FROM

  `total-scion-382610.target\_sql\_business\_case.orders`

  
Data is present from **2016-09-04 21:15:19 UTC** to **2018-10-17 17:30:18 UTC**

1.3 **Cities and States of customers ordered during the given period**

SELECT

  DISTINCT customer\_city,

  customer\_state

FROM

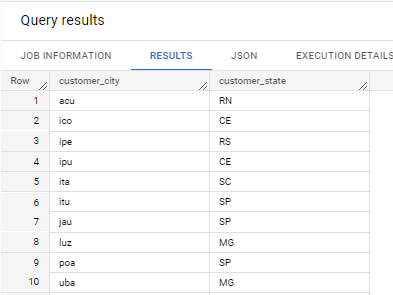
  `total-scion-382610.target\_sql\_business\_case.customers` AS cust

JOIN

  `total-scion-382610.target\_sql\_business\_case.orders` AS ord

ON

  cust.customer\_id = ord.customer\_id



**Question 2**

In-depth Exploration:

2.1 **Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?**

SELECT

  EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year,

  COUNT(DISTINCT order\_id) AS TotalOrders,

FROM

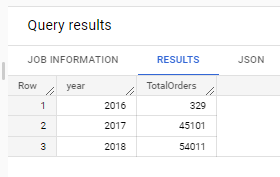
  `total-scion-382610.target\_sql\_business\_case.orders`

GROUP BY

  year

ORDER BY

  year ASC



As we can **notice year on year**, there has been an **increase** in the total number of orders placed which clearly indicates that there **has been a growing trend** of e-commerce in Brazil.

**Can we see some seasonality with peaks at specific months?**Not really, if we look at the below result set, in 2016 October we see peak number of orders as 324, while in 2017 peak orders is in the month of November and in 2018 it is in the month of March.

SELECT

  EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year,

  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

  COUNT(DISTINCT order\_id) AS TotalOrders,

  FROM

  `total-scion-382610.target\_sql\_business\_case.orders`

GROUP BY

  year,month

ORDER BY

  year,month ASC

|  |  |  |  |
| --- | --- | --- | --- |
| Row | year | month | TotalOrders |
| 1 | 2016 | 9 | 4 |
| 2 | 2016 | 10 | 324 |
| 3 | 2016 | 12 | 1 |
| 4 | 2017 | 1 | 800 |
| 5 | 2017 | 2 | 1780 |
| 6 | 2017 | 3 | 2682 |
| 7 | 2017 | 4 | 2404 |
| 8 | 2017 | 5 | 3700 |
| 9 | 2017 | 6 | 3245 |
| 10 | 2017 | 7 | 4026 |
| 11 | 2017 | 8 | 4331 |
| 12 | 2017 | 9 | 4285 |
| 13 | 2017 | 10 | 4631 |
| 14 | 2017 | 11 | 7544 |
| 15 | 2017 | 12 | 5673 |
| 16 | 2018 | 1 | 7269 |
| 17 | 2018 | 2 | 6728 |
| 18 | 2018 | 3 | 7211 |
| 19 | 2018 | 4 | 6939 |
| 20 | 2018 | 5 | 6873 |
| 21 | 2018 | 6 | 6167 |
| 22 | 2018 | 7 | 6292 |
| 23 | 2018 | 8 | 6512 |
| 24 | 2018 | 9 | 16 |
| 25 | 2018 | 10 | 4 |

**2.2 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?**I have considered below as the interval for Dawn, Morning, Afternoon, Night and as per the result set most orders are placed in afternoon followed by night by Brazilian customers.

|  |  |
| --- | --- |
| Dawn | 3:00:00-5:59:59 |
| Morning | 6:00:00-11:59:59 |
| Afternoon | 12:00:00-17:59:59 |
| Night | 18:00:00-2:59:59 |

SELECT

  COUNT(DISTINCT order\_id) AS TotalOrders,

  CASE

    WHEN h >= 3 AND h <= 5 AND m >=0 AND m <=59 AND s >=0 AND s <=59 THEN 'DAWN'

    WHEN h >= 6 AND h <= 11 AND m >=0 AND m <=59 AND s >=0 AND s <=59 THEN 'MORNING'

    WHEN h >= 12 AND h <= 17 AND m >=0 AND m <=59 AND s >=0 AND s <=59 THEN 'AFTERNOON'

    ELSE

    'NIGHT'

  END

    AS TIME

FROM

  (

    SELECT

      order\_id,

      EXTRACT(Hour FROM order\_purchase\_timestamp) AS h,

      EXTRACT(MINUTE FROM order\_purchase\_timestamp) AS m,

      EXTRACT(Second FROM order\_purchase\_timestamp) AS s

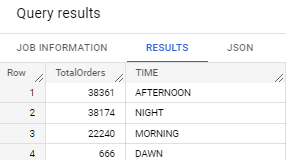
    FROM

      `total-scion-382610.target\_sql\_business\_case.orders`

  )

GROUP BY TIME

ORDER BY TotalOrders DESC

****

**Question 3**

Evolution of E-commerce orders in the Brazil region:

**3.1 Get month on month orders by states.**  
SELECT

  cust.customer\_state AS State,

  EXTRACT(MONTH FROM ord.order\_purchase\_timestamp) AS Month,

  COUNT(DISTINCT ord.order\_id) AS OrdersPerMonth

FROM

  `total-scion-382610.target\_sql\_business\_case.orders` AS ord

JOIN

  `total-scion-382610.target\_sql\_business\_case.customers` AS cust

ON

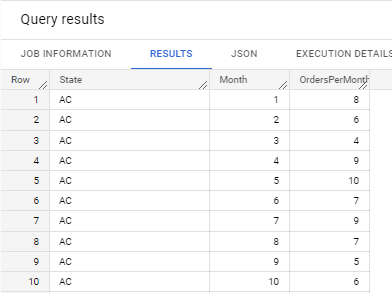
  ord.customer\_id = cust.customer\_id

GROUP BY

  cust.customer\_state, Month

ORDER BY

  cust.customer\_state, Month



**3.2 Distribution of customers across the states in Brazil**

SELECT

  cust.customer\_state AS State,

  COUNT(DISTINCT cust.**customer\_unique\_id**) AS CustomersInState

FROM

  `total-scion-382610.target\_sql\_business\_case.customers` AS cust

GROUP BY

  cust.customer\_state

  ORDER BY cust.customer\_state



Note : Have taken **customer\_unique\_id** for aggregation to find customers in a state.

**Question 4**

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

**4.1 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**SELECT

  R2.Year,

  R2.PreviousYearOrderValue,

  R2.CurrentYearOrderValue,

  IF

    (R2.PreviousYearOrderValue IS NOT NULL,ROUND((ABS(R2.CurrentYearOrderValue-R2.PreviousYearOrderValue)/R2.PreviousYearOrderValue)\*100,2),NULL) AS   PercentageIncreaseInOrderValue

FROM (

  SELECT

    R1.Year,

    LAG(R1.CurrentYearOrderValue,1) OVER (ORDER BY R1.YEAR) AS PreviousYearOrderValue,

    R1.CurrentYearOrderValue

  FROM

  (

        SELECT

          sub.year AS Year,

          ROUND(SUM(payment.payment\_value),2) AS CurrentYearOrderValue

        FROM

          `total-scion-382610.target\_sql\_business\_case.payments` AS payment

        JOIN

        (

            SELECT

              EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year,

              EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,

              order\_id

            FROM

              `total-scion-382610.target\_sql\_business\_case.orders`

        ) AS sub

        ON

          payment.order\_id = sub.order\_id

        WHERE

          sub.year IN (2017,2018) AND sub.month <= 8

        GROUP BY sub.YEAR

      ) AS R1

    )

    AS R2

ORDER BY

  R2.year

****

**4.2 Mean & Sum of price and freight value by customer state**

SELECT DISTINCT cust.customer\_state,

SUM(ordItems.freight\_value) over (partition by cust.customer\_state) AS SMFreightValue,

AVG(ordItems.freight\_value) over (partition by cust.customer\_state) AS MeanFreightValue,

SUM(ordItems.price) over (partition by cust.customer\_state) AS SMPrice,

AVG(ordItems.price) over (partition by cust.customer\_state) AS MeanPrice

FROM `target\_sql\_business\_case.order\_items` AS ordItems

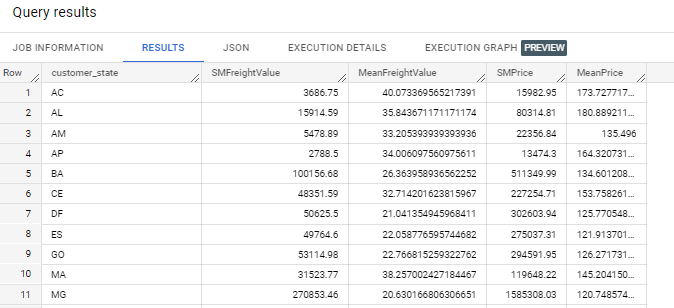
JOIN `target\_sql\_business\_case.orders` AS ord

ON ord.order\_id = ordItems.order\_id

JOIN `target\_sql\_business\_case.customers` AS cust

ON cust.customer\_id = ord.customer\_id

ORDER BY cust.customer\_state

**  
  
Question 5**Analysis on sales, freight, and delivery time.

**5.1 Calculate days between purchasing, delivering and estimated** **delivery**SELECT

  order\_purchase\_timestamp,

  order\_delivered\_customer\_date,

  order\_estimated\_delivery\_date,

IF

  (order\_purchase\_timestamp IS NOT NULL

    AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS DaysBtwnPurchaseAndDelivery,

IF

  (order\_purchase\_timestamp IS NOT NULL

    AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp,DAY), NULL) AS DaysBtwnPurchaseAndEstmDelivery,

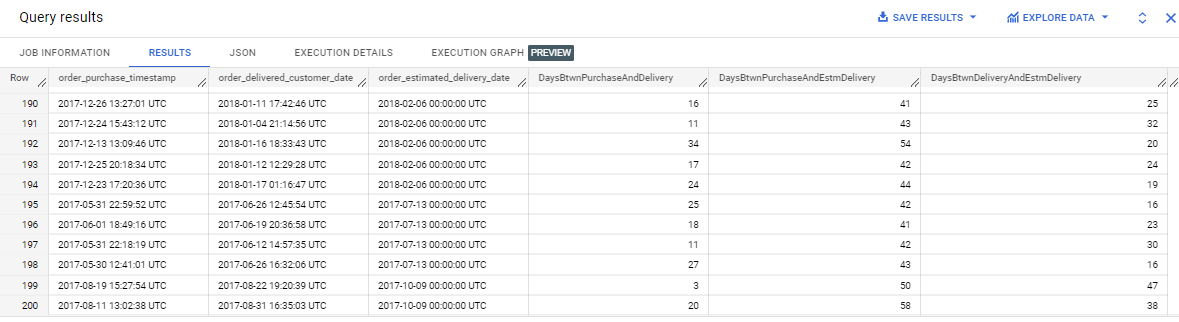
IF

  (order\_delivered\_customer\_date IS NOT NULL

    AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS DaysBtwnDeliveryAndEstmDelivery

FROM

  `target\_sql\_business\_case.orders`  
  
**NOTE** : Many of the **order\_delivered\_customer\_date** is null, during which **DaysBtwnPurchaseAndDelivery** will be null and **DaysBtwnDeliveryAndEstmDelivery** will also be null.

**  
  
  
5.2 Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:**

* **time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date**
* **diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date**

SELECT

IF

  (order\_purchase\_timestamp IS NOT NULL

    AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

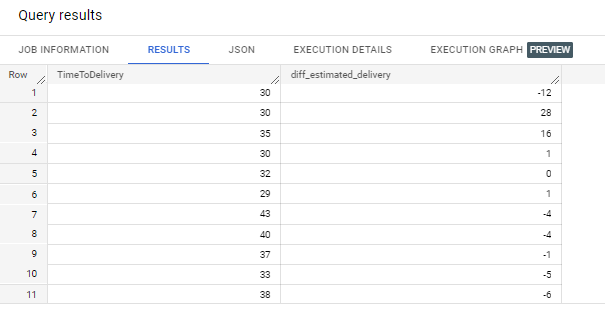
IF

  (order\_delivered\_customer\_date IS NOT NULL

    AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

FROM

  `target\_sql\_business\_case.orders`

****

**Negative diff\_estimated\_delivery** means customer got the product at a date later than the Estimated delivery date.   
  
**5.3 Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery**

SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

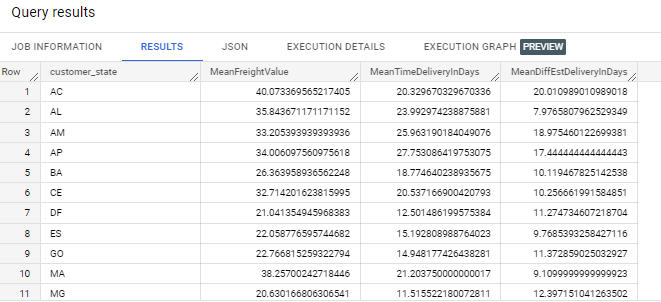
  ord.order\_id = ordItems.order\_id

GROUP BY

  cust.customer\_state

ORDER BY

  cust.customer\_state

****

**5.4 Sort the data to get the following:**

**5.5.1** Top 5 states with **highest** average freight value - sort in desc limit 5SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

  ord.order\_id = ordItems.order\_id

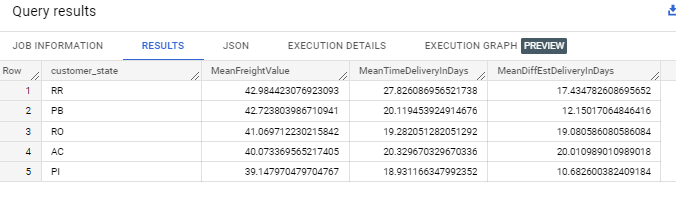
GROUP BY

  cust.customer\_state

**ORDER BY**

**MeanFreightValue DESC**

**LIMIT 5**

****

**5.5.2** Top 5 states with **lowest** average freight value - sort in asc limit 5  
  
SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

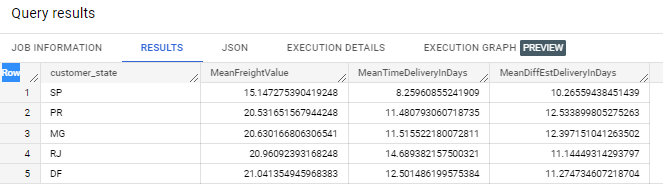
  ord.order\_id = ordItems.order\_id

GROUP BY

  cust.customer\_state

ORDER BY

  MeanFreightValue ASC

LIMIT 5    
  
****

* + 1. **Top 5 states with highest average time to delivery**SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeToDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

  ord.order\_id = ordItems.order\_id

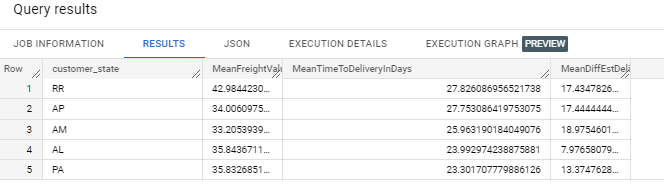
GROUP BY

  cust.customer\_state

ORDER BY

  MeanTimeToDeliveryInDays DESC

LIMIT 5



* + 1. **Top 5 states with lowest average time to delivery**

SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeToDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

  ord.order\_id = ordItems.order\_id

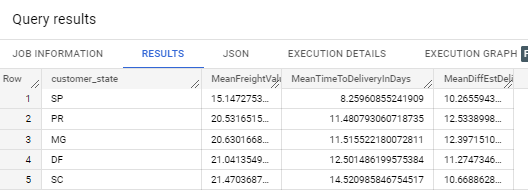
GROUP BY

  cust.customer\_state

ORDER BY

  MeanTimeToDeliveryInDays ASC

LIMIT 5

****

* + 1. **Top 5 states where delivery is really fast compared to estimated date**

SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeToDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

  ord.order\_id = ordItems.order\_id

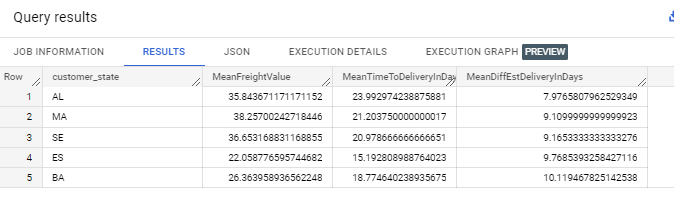
GROUP BY

  cust.customer\_state

ORDER BY

  MeanDiffEstDeliveryInDays ASC

LIMIT 5

****

* + 1. **Top 5 states where delivery is not so fast compared to estimated date**SELECT

  cust.customer\_state,

  AVG(ordItems.freight\_value) AS MeanFreightValue,

  AVG(ord.TimeToDelivery) AS MeanTimeToDeliveryInDays,

  AVG(ord.diff\_estimated\_delivery) AS MeanDiffEstDeliveryInDays

FROM

  `target\_sql\_business\_case.customers` AS cust

JOIN (

  SELECT

    customer\_id,

    order\_id,

  IF

    (order\_purchase\_timestamp IS NOT NULL

      AND order\_delivered\_customer\_date IS NOT NULL, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY), NULL) AS TimeToDelivery,

  IF

    (order\_delivered\_customer\_date IS NOT NULL

      AND order\_estimated\_delivery\_date IS NOT NULL, DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY), NULL) AS diff\_estimated\_delivery

  FROM

    `target\_sql\_business\_case.orders`) AS ord

ON

  cust.customer\_id = ord.customer\_id

JOIN

  `target\_sql\_business\_case.order\_items` AS ordItems

ON

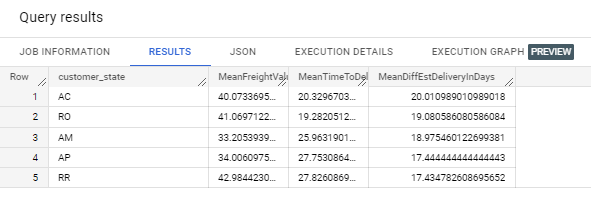
  ord.order\_id = ordItems.order\_id

GROUP BY

  cust.customer\_state

ORDER BY

  MeanDiffEstDeliveryInDays DESC

LIMIT 5    
  
****

**Question 6  
  
Payment type analysis:**

**6.1 Month over Month count of orders for different payment types**SELECT

  ordTemp.Year,

  ordTemp.Month,

  pay.payment\_type AS PaymentType,

  COUNT(DISTINCT ord.order\_id) AS OrdersCount

FROM

  `target\_sql\_business\_case.orders` AS ord

JOIN

  `target\_sql\_business\_case.payments` AS pay

ON

  ord.order\_id = pay.order\_id

JOIN (

  SELECT

    EXTRACT(Year

    FROM

      order\_purchase\_timestamp) AS Year,

    EXTRACT(Month

    FROM

      order\_purchase\_timestamp) AS Month,

    order\_id

  FROM

    `total-scion-382610.target\_sql\_business\_case.orders` ) AS ordTemp

ON

  ord.order\_id = ordTemp.order\_id

GROUP BY

  ordTemp.Year,

  ordTemp.Month,

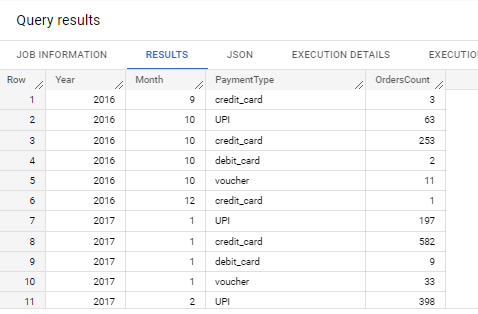
  pay.payment\_type

ORDER BY

 ordTemp.Year,

  ordTemp.Month,

  pay.payment\_type

****

**6.2 Count of orders based on the no. of payment instalments**SELECT

  pay.payment\_installments AS PaymentInstallments,

  COUNT(DISTINCT ord.order\_id) AS OrdersCount

FROM

  `target\_sql\_business\_case.orders` AS ord

JOIN

  `target\_sql\_business\_case.payments` AS pay

ON

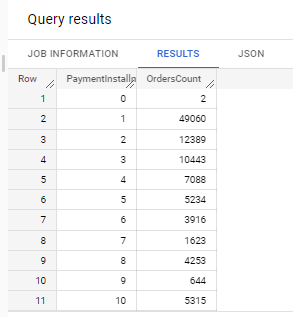
  ord.order\_id = pay.order\_id

GROUP BY

  pay.payment\_installments

ORDER BY

  pay.payment\_installments

****

**Question-7 Actionable Insights**

**1. States** RR, AP, AC, AM, RO has lowest customers count for target. Some Promotional offers/Marketing Campaigns could be run in these states to attract customer traction.

SELECT

  cust.customer\_state AS State,

  COUNT(DISTINCT cust.customer\_unique\_id) AS CustomersInState

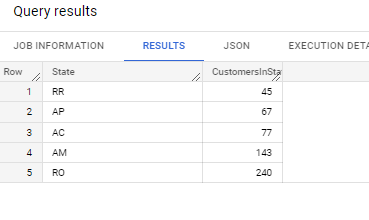
FROM

  `total-scion-382610.target\_sql\_business\_case.customers` AS cust

GROUP BY

  cust.customer\_state

  ORDER BY CustomersInState ASC LIMIT 5

****

**Question 8 Recommendations  
  
1. As we** saw as part of Question 2, Brazilian customers tend to buy most in Afternoon followed by Night, offers/promotions should be done in this window to attract more customers and increasing the total orders/revenue.  
  
**2. Mostly** customers are paying in full for the orders as OrdersCount is MAX where PaymentInstallments is 1. Promotional Offers for EMIs /Credit Card could be run so that customers are able to buy more orders of their choice.  
  
**Count of orders based on the no. of payment instalments**SELECT

  pay.payment\_installments AS PaymentInstallments,

  COUNT(DISTINCT ord.order\_id) AS OrdersCount

FROM

  `target\_sql\_business\_case.orders` AS ord

JOIN

  `target\_sql\_business\_case.payments` AS pay

ON

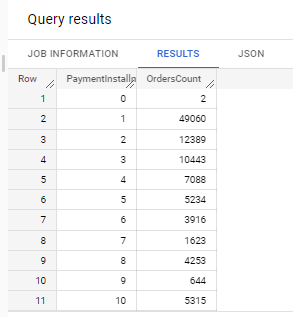
  ord.order\_id = pay.order\_id

GROUP BY

  pay.payment\_installments

ORDER BY

  pay.payment\_installments

****