**Business Case: Target SQL**



Context

Target is one of the world’s most recognized brands and one of America’s leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

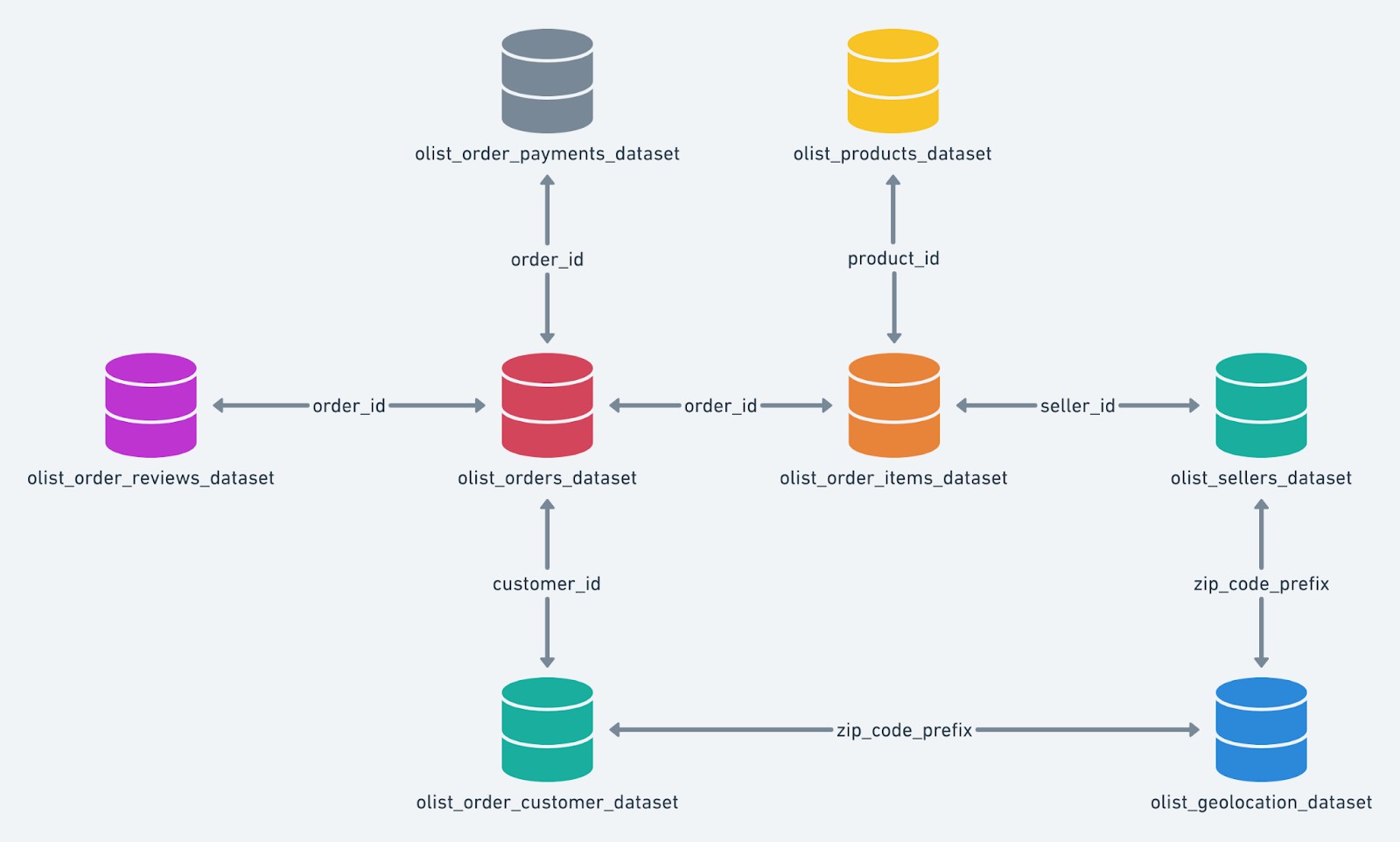
This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and ﬁnally reviews written by customers.

Data is available in 8 csv ﬁles:

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

Each feature or columns of different CSV ﬁles are described below:

High level overview of relationship between datasets:



Assume you are a data scientist at Target, and are given this data to analyze and provide some insights and recommendations from it.

* 1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
     1. Data type of columns in a table
     2. Time period for which the data is given
     3. Cities and States of customers ordered during the given period
  2. In-depth Exploration:
     1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at speciﬁc months?
     2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
  3. Evolution of E-commerce orders in the Brazil region:
     + 1. Get month on month orders by states
       2. Distribution of customers across the states in Brazil
  4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
     + 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
       2. Mean & Sum of price and freight value by customer state
  5. Analysis on sales, freight and delivery time
     + 1. Calculate days between purchasing, delivering and estimated delivery
       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
       3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery
       4. Sort the data to get the following:
       5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
       6. Top 5 states with highest/lowest average time to delivery
       7. Top 5 states where delivery is really fast/ not so fast compared to estimated date
     1. Payment type analysis:
        1. Month over Month count of orders for different payment types
        2. Count of orders based on the no. of payment installments

# SOLUTION:

|  |
| --- |
| Q1)Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.   1. Data type of columns in a table 2. Time period for which the data is given 3. Cities and States of customers ordered during the given period |
| 1.1)Data type of columns in a table |
| **Customers Table:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.customers` LIMIT 1000 |
|  |
| **GeoLocation Table:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.geolocation` LIMIT 1000 |
|  |

|  |
| --- |
| **Order Items Table:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.order\_items` LIMIT 1000 |
|  |
| **Orders Table:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.orders` LIMIT 1000 |
|  |
| **Payments:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.payments` LIMIT 1000 |
|  |

|  |
| --- |
| **Products:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.products` LIMIT 1000 |
|  |
| **Sellers:**  SELECT \* FROM `business-case-study-sql.Target\_dataset.sellers` LIMIT 1000 |
|  |

|  |
| --- |
| 1.1) Data type of columns in a table |
| **Customers table:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'customers' |
|  |
| **GeoLocation:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'geolocation' |
|  |

|  |
| --- |
| **Order items:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'order\_items' |
|  |
| **Orders:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'orders' |
|  |
|  |

|  |
| --- |
| **Payments:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'payments' |
|  |

|  |
| --- |
| **Products:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'products' |
|  |
|  |

|  |
| --- |
| **Sellers:**  SELECT column\_name, data\_type  FROM `business-case-study-sql.Target\_dataset.INFORMATION\_SCHEMA.COLUMNS` WHERE table\_name = 'sellers' |
|  |

|  |
| --- |
| 1.2)Time period for which the data is given |
| SELECT  MIN(order\_purchase\_timestamp) AS start\_date, MAX(order\_purchase\_timestamp) AS end\_date  FROM `business-case-study-sql.Target\_dataset.orders` |
|  |
| Insights:  Start date = 04-09-2016 End date = 17-10-2018  Time period = 2 years, 7 months, 14 days including the end date. |

|  |
| --- |
| 1.3)Cities and States of customers ordered during the given period |
| SELECT DISTINCT customer\_city, customer\_state  FROM `business-case-study-sql.Target\_dataset.customers` AS c JOIN `business-case-study-sql.Target\_dataset.orders` AS o ON c.customer\_id = o.customer\_id |
|  |
| Insights:  There are total 4311 rows. Count of unique cities = 4119 Count of unique states = 27 |

|  |
| --- |
| SELECT DISTINCT geolocation\_city, geolocation\_state  FROM `business-case-study-sql.Target\_dataset.geolocation` |
|  |
|  |

|  |
| --- |
| SELECT DISTINCT customer\_city, customer\_state  FROM `business-case-study-sql.Target\_dataset.customers` |
|  |
| Insights:  There are total 8464 rows. Count of unique cities = 4119 Count od unique states = 27 |

|  |
| --- |
| Q2) In-depth Exploration:  2.1)Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?  Can we see some seasonality with peaks at speciﬁc months? |
| SELECT  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS order\_month, EXTRACT(YEAR FROM order\_purchase\_timestamp) AS order\_year, COUNT(DISTINCT order\_id) AS order\_count, SUM(DISTINCT(order\_id)) AS total\_sales  FROM `business-case-study-sql.Target\_dataset.orders` GROUP BY order\_month, order\_year  ORDER BY order\_year, order\_month |
|  |
|  |

|  |
| --- |
| Insights:   1. We can clearly see that that there a growing trend on e-commerce in Brazil from 2016 to 2018. 2. Nov 2017 is the month when most of the orders were placed. 3. Outlier data points: (09 &12 )in 2016 and (09 & 10) in 2018   Here, the number of order\_count is very less than the median data. It seems the data was not captured correctly and looks corrupted. If these datapoints are not corrected then it may negatively impact the analysis of the entire dataset. |

|  |
| --- |
| SELECT  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS order\_month, EXTRACT(YEAR FROM order\_purchase\_timestamp) AS order\_year, COUNT(DISTINCT c.order\_id) AS order\_count, SUM(p.payment\_value) AS total\_sales  FROM `business-case-study-sql.Target\_dataset.orders` AS c JOIN `business-case-study-sql.Target\_dataset.payments` AS p ON c.order\_id = p.order\_id  GROUP BY  order\_month, order\_year ORDER BY  order\_year, order\_month |
|  |
|  |

|  |
| --- |
|  |
|  |
| Insights:   1. Seasonality chart (Month wise date added from Sept 2016 to Oct 2018)    1. August month has the highest number of orders. (Maybe there is Big Sale Offer/Stock Clearance offer) 1.2)September month has the lowest number of orders. 2. The average number of orders for the ﬁrst 8 months is higher than last 4 months. Outlier data points: (09 &12 )in 2016 and (09 & 10) in 2018   Here, the number of order\_count is very less than the median data. It seems the data was not captured correctly and looks corrupted. If these datapoints are not corrected then it may negatively impact the analysis of the entire dataset.   1. Since Brazil lies in South Hemisphere, summer is December through March and winter June through September. So, we observe that the number of orders are more in winter(June to September) than summer season(December to March).   Therefore, we observe that the number of orders depends on the seasons. |

|  |
| --- |
| 2.2) What time do Brazilian customers tend to buy  (Dawn - 12am-6am, Morning 6am-12pm, Afternoon - 12 noon to 6pm,or Night 6-pm - 12am)? |
| SELECT CASE  WHEN EXTRACT(hour FROM timestamp(order\_purchase\_timestamp)) BETWEEN 0 AND 6 THEN 'dawn'  WHEN EXTRACT(hour FROM timestamp(order\_purchase\_timestamp)) BETWEEN 7 AND 12 THEN 'morning'  WHEN  EXTRACT(hour FROM timestamp(order\_purchase\_timestamp)) BETWEEN 13 AND 18 THEN 'afternoon'  WHEN  EXTRACT(hour FROM timestamp(order\_purchase\_timestamp)) BETWEEN 19 AND 23 THEN 'night'  END AS time\_of\_day,  COUNT(DISTINCT order\_id) AS counter  FROM `business-case-study-sql.Target\_dataset.orders` GROUP BY 1  ORDER BY 2 DESC; |
|  |
| Insights:  Here the time slots are:  Dawn:- 12am-6am, Morning:- 7am-12pm, Afternoon:- 1pm to 6pm, Night:- 7pm - 11pm |
| SELECT CASE  WHEN TIME(order\_purchase\_timestamp) BETWEEN '00:00:00' AND '06:59:59' THEN 'Dawn'  WHEN TIME(order\_purchase\_timestamp) BETWEEN '07:00:00' AND '12:59:59' THEN 'Morning'  WHEN TIME(order\_purchase\_timestamp) BETWEEN '13:00:00' AND '18:59:59' THEN 'Afternoon'  ELSE 'Night'  END AS TIME\_OF\_PURCHASE,  COUNT(DISTINCT order\_id) AS num\_orders  FROM `business-case-study-sql.Target\_dataset.orders` GROUP BY TIME\_OF\_PURCHASE  order by num\_orders |

|  |
| --- |
|  |
| Here the time slots are:  Dawn - 12:00:00 am-6:59:59 am, Morning 07:00:00 am-12:59:59 pm, Afternoon - 01:00:00 pm to 6:59:59 pm, or Night 07:00:00 pm - 11:59:59 pm |
| SELECT CASE  WHEN TIME(order\_purchase\_timestamp) BETWEEN '00:00:00' AND '05:59:59' THEN 'Dawn' WHEN TIME(order\_purchase\_timestamp) BETWEEN '06:00:00' AND '11:59:59' THEN 'Morning'  WHEN TIME(order\_purchase\_timestamp) BETWEEN '12:00:00' AND '17:59:59' THEN 'Afternoon' ELSE 'Night'  END AS TIME\_OF\_PURCHASE,  COUNT(DISTINCT order\_id) AS num\_orders  FROM `business-case-study-sql.Target\_dataset.orders` GROUP BY TIME\_OF\_PURCHASE  order by num\_orders |
|  |
| Here the time slots are:  Dawn - 12:00:00 am-5:59:59 am, Morning 06:00:00 am-11:59:59 pm, Afternoon - 12:00:00 pm to 5:59:59 pm, or Night 06:00:00 pm - 11:59:59 pm |
| Insights:   1. During the afternoon time (12pm-6pm), most of the customers place order. Reason: People are awake during this time period.   So, most advertisements can be shown during this time period.   1. During morning and night, almost equal amount of orders are placed. 2. During dawn timing(12am-6am), very less people order. Reason: People are mostly asleep during this time. |

|  |
| --- |
| Q3)Evolution of E-commerce orders in the Brazil region:  3.1)Get month on month orders by states |
| SELECT  EXTRACT(month FROM timestamp(order\_purchase\_timestamp)) AS month, g.geolocation\_state,  COUNT(1) AS num\_orders  FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` c ON o.customer\_id = c.customer\_id  INNER JOIN `business-case-study-sql.Target\_dataset.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix GROUP BY g.geolocation\_state, month  ORDER BY geolocation\_state DESC, month ASC |
|  |
| Sheets Plot: |
| Most of the orders are from AL state.  Here, is the month vs num\_orders plot for AL state. |

|  |
| --- |
| Least of the orders are from SE state.  Here, is the month vs num\_orders plot for SE state. |
| Tableau visualization: |
|  |

|  |
| --- |
| 3.2)Distribution of customers across the states in Brazil  SELECT g.geolocation\_state, COUNT(DISTINCT (c.customer\_unique\_id)) AS num\_customers FROM `business-case-study-sql.Target\_dataset.customers` c  INNER JOIN `business-case-study-sql.Target\_dataset.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix GROUP BY g.geolocation\_state  ORDER BY num\_customers DESC; |
|  |
| Sheets plot: |
|  |

|  |
| --- |
| Tableau visualization: |
|  |
| Insights:  The Top most number of orders is placed from the SP state. The 2nd most number of orders is placed from the RJ state. The 3rd most number of orders is placed from the MG state.  We should further analyse the main reasons for the most number of orders. Can we increase the number of orders even more ? If yes, then how?  The least number of orders is placed from the RR state.  We should further analyse the main reasons for the least number of orders . What can we do to increase the number of orders?  We observe that:  The state SP has the maximum number of customers & the maximum number of orders. Hence, the maximum number of sales.  The state GO has very less number of customers but has very large amount of order. |

|  |
| --- |
| Q4)Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others (*charges*).  4.1)Get % increase in cost of orders (*refer to payment\_value column in payment table*) from **2017 to 2018** (include months between **Jan to Aug only**(*month between 1 to 8*)) - You can use “payment\_value” column in payments table |
| Approach:  Create CTE Table and new columns: price\_per\_order = sum(price)/count(order\_id)  freight\_per\_order= sum(freight\_value)/count(order\_id) Group the data on yearly and monthly level |
| WITH  cte\_table AS ( SELECT  EXTRACT(month FROM timestamp(o.order\_purchase\_timestamp)) AS month, EXTRACT(year FROM timestamp(o.order\_purchase\_timestamp)) AS year, (sum(price) / COUNT( distinct o.order\_id)) AS price\_per\_order, (sum(freight\_value) / COUNT(distinct o.order\_id)) AS freight\_per\_order FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.order\_items` i ON o.order\_id = i.order\_id  GROUP BY year, month  )  SELECT (price\_per\_order), (freight\_per\_order), month, year FROM cte\_table  order by year asc, month asc ; |
|  |
| Sheets Plot: |

|  |
| --- |
|  |
|  |
| Insights:  We observe that:  For Year wise plot   1. For price\_per\_order   From 2016 to 2017:- there is signiﬁcant growth of From 2017 to 2018:- there is slight decrease of   1. For freight\_per\_order   From 2016 to 2017:- there is increase of  From 2017 to 2018:- there is slight decrease of  For month on month plot:  For Year wise plot   1. For price\_per\_order September is the highest   November & December is the lowest   1. For freight\_per\_order September is the highest   November & December is the lowest  Note:  Freight transport, also referred as freight forwarding, is the physical process of transporting commodities and merchandise goods and cargo.  A freight rate is a price at which a certain cargo is delivered from one point to another. |

|  |
| --- |
| The price depends on the form of the cargo, the mode of transport (truck, ship, train, aircraft), the weight of the cargo, and the distance to the delivery destination.  Here,  price\_per\_order = freight\_per\_order + other charges Therefore,  other charges = price\_per\_order - freight\_per\_order |

|  |
| --- |
| Total amount sold in 2017 between Jan to August (Jan to Aug because data is available starting 2017 01 to 2018 08) and we can only compare cycles with cycles  Compare YoY at a monthly level |
| WITH  cte\_table AS ( SELECT  EXTRACT(month FROM timestamp(order\_purchase\_timestamp)) AS month, EXTRACT(year FROM timestamp(order\_purchase\_timestamp)) AS year, sum(price) AS total\_price,  sum(freight\_value) AS total\_freight  FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.order\_items` i ON o.order\_id = i.order\_id  GROUP BY year, month  ORDER BY year ASC, month ASC  ) SELECT  month, price\_2017, price\_2018,  round((price\_2018 - price\_2017) / price\_2017 \* 100, 2) AS yoy FROM  (  SELECT  month,  sum(CASE WHEN year = 2017 THEN total\_price ELSE 0 END) AS price\_2017, sum(CASE WHEN year = 2018 THEN total\_price ELSE 0 END) AS price\_2018  -- sum(total\_price) as total\_transaction\_amt FROM cte\_table  WHERE (year = 2017 OR year = 2018) AND month BETWEEN 1 AND 8  GROUP BY month order by month  ); |

|  |
| --- |
|  |
| Sheets plot: |
| Insights:  We observe that:  For YoY at amonthly level (from 2017 to 2018).  For January, the Total amount sold change is maximum with ~690% increase. This is a huge growth. What are the main factors driving this massive growth?  In general, for all months there is increase in the Total amount sold (from 2017 to 2018). This means that Brazil is a growing economy for e-commerce market. |
| MoM increase for year 2017 |
|  |

|  |
| --- |
| SELECT  month, orders, lagger\_orders,  (orders - coalesce(lagger\_orders, 0)) / coalesce(lagger\_orders, 1) \* 100 AS difference FROM  (  SELECT \*, lag(orders, 1) OVER (ORDER BY month ASC) AS lagger\_orders FROM  (  SELECT  EXTRACT(month FROM timestamp(a.order\_purchase\_timestamp)) AS month, COUNT(DISTINCT a.order\_id) AS orders,  COUNT(DISTINCT b.customer\_unique\_id) AS customers  FROM `business-case-study-sql.Target\_dataset.orders` a  LEFT JOIN `business-case-study-sql.Target\_dataset.customers` b ON a.customer\_id = b.customer\_id  WHERE EXTRACT(year FROM timestamp(a.order\_purchase\_timestamp)) = 2017 GROUP BY 1  ) base  ) base\_2  ORDER BY month ASC; |
|  |

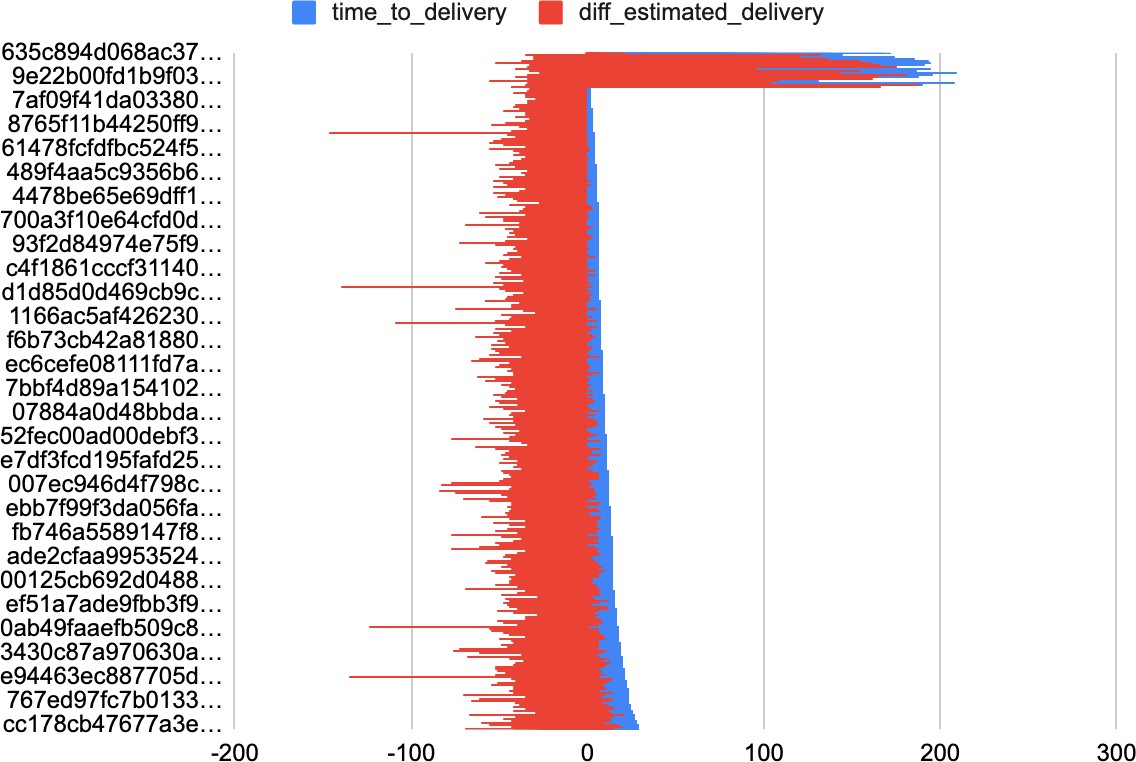
|  |
| --- |
|  |
| Insights:  We observe that:  For the year 2017, there is month on month increase in Total sales for all the months. Moreover, the amount of increase per month is also increasing across the month (in general). |

|  |
| --- |
| 4.2)Mean(avg) & Sum of price and freight value by customer state |
| Sum and mean price by customer state  It's very interesting to see how some states have a high total amount sold and a low price per order.  If we look at SP (São Paulo) for example, it's possible to see that it  is the state with most valuable state for e-commerce (5202955 sold) but it is also where customers pay less per order (125.75 per order) |
| with cte\_table as ( select  c.customer\_state as state, sum(price) as total\_price,  count(distinct(o.order\_id)) as num\_orders  from `business-case-study-sql.Target\_dataset.orders` o  inner join `business-case-study-sql.Target\_dataset.order\_items` i on o.order\_id= i.order\_id  inner join `business-case-study-sql.Target\_dataset.customers` c on o.customer\_id=c.customer\_id  group by state  )  select state, total\_price, num\_orders,(total\_price/num\_orders) as avg\_price  from cte\_table  order by total\_price desc; |

|  |
| --- |
|  |
| Sheets plot: |
|  |

|  |
| --- |
|  |
| Insights:  We observe that:  1)For state wise total price SP is the highest  RR is the lowest   1. For state wise avg price PB is the highest   SP is the lowest   1. For state wise num\_orders SP is the highest   RR is the lowest  So, for SP num\_orders is the highest & avg\_price is lowest. Hence, Total price is maximum.  This means customers prefer more number of lower price products. |

|  |
| --- |
| Q5) Analysis on sales, freight and delivery time   1. **Calculate days between purchasing, delivering and estimated delivery:**   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  1. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery   **Sort the data to get the following:**   1. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5 2. Top 5 states with highest/lowest average time to delivery 3. Top 5 states where delivery is really fast/ not so fast compared to estimated date |
| 5.1)Calculate days between purchasing, delivering and estimated delivery: |
| Approach:  create new columns for time to delivery and difference in estimated vs actual delivery |
| SELECT  order\_id, date\_DIFF(date(order\_delivered\_customer\_date),date(order\_purchase\_timestamp),DAY) AS time\_to\_delivery, date\_DIFF(date(order\_delivered\_customer\_date),date(order\_estimated\_delivery\_date),DAY) AS diff\_estimated\_delivery  FROM `business-case-study-sql.Target\_dataset.orders` WHERE order\_status = 'delivered'; |
|  |
| Sheets plot: |



Using basic statistics to get the insights from data:

|  |  |  |
| --- | --- | --- |
|  | Time\_to\_delivery (days) | Diff\_estimated\_delivery (days) |
| Maximum | 210 | 188 |
| Minimum | 0 | -56 |
| Average | 12.49 | -11.87 |
| Median | 10 | -6 |
| 90th percentile | 23 | -2 |
| 95th percentile | 29 | 3 |
| 99th percentile | 83 | 18 |
| 99.9th percentile | 83 | 55 |
| 99.99th percentile | 187.35 | 156.41 |

|  |
| --- |
| Insights:  We can clearly observe the maximum, minimum, average, median time to delivery & difference in estimated vs actual delivery (in days).  According to Median value, for delivery it takes 10 days.  And difference in estimated vs actual delivery (in days) is 6 days before actual delivery date.  The delivery team shold work on the 95th percentile (~29 days) to 99.9th percentile(~83 days) range. The logistics team should identify the reason for such huge delay in delivery time (29 days-83 days) and Find relevant ways to reduce this huge delivery time.  The delivery team shold work on the 99th percentile (~18 days) to 99.9th percentile(~55 days) range.  The logistics team should identify the reason for such huge delay in difference in estimated vs actual delivery time (18 days-55 days) and  Find relevant ways to reduce this huge delivery time.  The maximum delivery time(~210 days) can be considered as an outlier.  The maximum difference in estimated vs actual delivery time(~188 days) can be considered as an outlier.  Faster delivery will result in customer satisfaction. This leads to more number of orders per customers. Therefore, the total sales will increase. |

|  |
| --- |
| 5.2) Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery |
| SELECT  c.customer\_state AS state, AVG(it.freight\_value) AS avg\_freight\_value,  AVG(date\_DIFF(date(o.order\_delivered\_customer\_date), date(o.order\_purchase\_timestamp), DAY)) AS avg\_time\_to\_delivery,  AVG(date\_DIFF(date(o.order\_estimated\_delivery\_date), date(o.order\_delivered\_customer\_date), DAY)) AS avg\_diff\_estimated\_delivery  FROM `business-case-study-sql.Target\_dataset.orders` AS o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` AS c ON o.customer\_id = c.customer\_id  INNER JOIN `business-case-study-sql.Target\_dataset.order\_items` AS it ON o.order\_id = it.order\_id  WHERE order\_status = 'delivered' GROUP BY state; |

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
|  |
| Insights:  We observe that:   1. For state wise avg freight value Maximum value = state RR Minimum value = state SP 2. For state wise avg time to delivery Maximum value = RR (~28 days) Minimum value = SP (~8 days) 3. For state wise difference in estimated vs actual delivery time Maximum value = AC (~22 days)   Minimum value = AL (~8 days)  Therefore, we can infer that  due to maximum avg freight value and maximum avg time to delivery time the state RR has lowest sales. due to minimum avg freight value and minimum avg time to delivery time the state SP has lowest sales.  Hence, to increase the total sales the avg freight value shold be minimized and avg time to delivery time should be minimized. |

|  |
| --- |
| 5.3) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5 |
| --ORDER BY avg\_freight\_value DESC SELECT  c.customer\_state AS state, AVG(it.freight\_value) AS avg\_freight\_value,  AVG(date\_DIFF(date(o.order\_delivered\_customer\_date), date(o.order\_purchase\_timestamp), DAY)) AS avg\_time\_to\_delivery,  AVG(date\_DIFF(date(o.order\_estimated\_delivery\_date), date(o.order\_delivered\_customer\_date), DAY)) AS avg\_diff\_estimated\_delivery  FROM `business-case-study-sql.Target\_dataset.orders` AS o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` AS c ON o.customer\_id = c.customer\_id  INNER JOIN `business-case-study-sql.Target\_dataset.order\_items` AS it ON o.order\_id = it.order\_id  WHERE order\_status = 'delivered' GROUP BY state  ORDER BY avg\_freight\_value DESC LIMIT 5; |
|  |
| Sheets plot: |
|  |

|  |
| --- |
|  |
| Insights:  We observe that:  Maximum avg freight value = RR & PB state (~42 units) Maximum avg time to delivery = RR state (~28 days)  Maximum difference in estimated vs actual delivery = AC state (~21 days) |

|  |
| --- |
| --ORDER BY avg\_freight\_value ASC SELECT  c.customer\_state AS state, AVG(it.freight\_value) AS avg\_freight\_value,  AVG(date\_DIFF(date(o.order\_delivered\_customer\_date), date(o.order\_purchase\_timestamp), DAY)) AS avg\_time\_to\_delivery,  AVG(date\_DIFF(date(o.order\_estimated\_delivery\_date), date(o.order\_delivered\_customer\_date), DAY)) AS avg\_diff\_estimated\_delivery  FROM `business-case-study-sql.Target\_dataset.orders` AS o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` AS c ON o.customer\_id = c.customer\_id  INNER JOIN `business-case-study-sql.Target\_dataset.order\_items` AS it ON o.order\_id = it.order\_id  WHERE order\_status = 'delivered' GROUP BY state  ORDER BY avg\_freight\_value ASC LIMIT 5; |

|  |
| --- |
|  |
| Sheets Plot: |
|  |
| Insights:  We observe that:  Minimum avg freight value = SP state (~15 units) Minimum avg time to delivery = SP state (~8 days)  Minimum difference in estimated vs actual delivery = SP state (~12 days) |

|  |
| --- |
| 5.4) Top 5 states with highest/lowest average time to delivery |
| --highest delivery time SELECT  g.geolocation\_state AS state,  SUM(TIMESTAMP\_DIFF(TIMESTAMP(order\_estimated\_delivery\_date), TIMESTAMP(order\_purchase\_timestamp), DAY)) / COUNT(ORDER\_ID) AS avg\_del\_time  FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` c ON o.customer\_id = c.customer\_id INNER JOIN `business-case-study-sql.Target\_dataset.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix  WHERE order\_status = 'delivered' GROUP BY g.geolocation\_state ORDER BY avg\_del\_time DESC |
|  |
| Sheets Plot: |
|  |
| Insights:  We observe that:  The highest delivery time = AP, RR, AM state (~46 days) |

|  |
| --- |
|  |
| --lowest delivery time  SELECT  g.geolocation\_state AS state,  SUM(TIMESTAMP\_DIFF(TIMESTAMP(order\_estimated\_delivery\_date), TIMESTAMP(order\_purchase\_timestamp), DAY)) / COUNT(ORDER\_ID) AS avg\_del\_time  FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` c ON o.customer\_id = c.customer\_id INNER JOIN `business-case-study-sql.Target\_dataset.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix  WHERE order\_status = 'delivered' GROUP BY g.geolocation\_state ORDER BY avg\_del\_time ASC |
|  |
| Sheets Plot: |
|  |

|  |
| --- |
| Insights:  We observe that regarding the delivery time that SP state is delivering the fastest (~19 days) This may be the reason for people ordering the most this state (as already analysed in 4.2). The slowest delivery time is for AP state (~46 days).  The customers will lose interest to order because of this huge delay in delivery. The delivery partners need to ﬁnd some ways to decrease the delivery time. |

|  |
| --- |
| 5.5) Top 5 states where delivery is really fast/ not so fast compared to estimated date |
| --Top 5 states where delivery is really fast SELECT  g.geolocation\_state AS state, AVG(TIMESTAMP\_DIFF(TIMESTAMP(order\_estimated\_delivery\_date),  TIMESTAMP(order\_delivered\_customer\_date), DAY)) AS diff\_estimated\_delivery FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` c ON o.customer\_id = c.customer\_id INNER JOIN `business-case-study-sql.Target\_dataset.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix  WHERE order\_status = 'delivered' GROUP BY g.geolocation\_state  ORDER BY diff\_estimated\_delivery ASC LIMIT 5; |
|  |
| Sheets Plot: |
|  |
| Insights:  We observe that  The fastest delivery is for AL state (~8 days). |

|  |
| --- |
| --Top 5 states where delivery is not so fast compared to estimated date SELECT  g.geolocation\_state AS state,  AVG(TIMESTAMP\_DIFF(TIMESTAMP(order\_estimated\_delivery\_date), TIMESTAMP(order\_delivered\_customer\_date), DAY)) AS diff\_estimated\_delivery FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.customers` c ON o.customer\_id = c.customer\_id INNER JOIN `business-case-study-sql.Target\_dataset.geolocation` g ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix  WHERE order\_status = 'delivered' GROUP BY g.geolocation\_state  ORDER BY diff\_estimated\_delivery DESC LIMIT 5; |
|  |
| Sheets Plot: |

|  |
| --- |
|  |
| Insights:  We observe that  The slowest delivery is for RR state (~21days). |

|  |
| --- |
| Q6) Payment type analysis:   1. Month over Month **count of orders** for different payment types 2. Count of orders based on the no. of payment installments |
| 6.1)Month over Month **count of orders** for different payment types |
| WITH  cte\_table AS ( SELECT  EXTRACT(month FROM timestamp(o.order\_purchase\_timestamp)) AS month, EXTRACT(year FROM timestamp(o.order\_purchase\_timestamp)) AS year, (sum(price) / COUNT( distinct o.order\_id)) AS price\_per\_order, (sum(freight\_value) / COUNT(distinct o.order\_id)) AS freight\_per\_order FROM `business-case-study-sql.Target\_dataset.orders` o  INNER JOIN `business-case-study-sql.Target\_dataset.order\_items` i ON o.order\_id = i.order\_id  GROUP BY year, month  )  SELECT (price\_per\_order), (freight\_per\_order), month, year FROM cte\_table  order by payment\_type, year asc, month asc ; |
|  |
| Sheets Plot: |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
| Insights:  We observe that most of the payments are done using Credit Card. The second most popular method of payment is UPI.  Then vouchers are used for payments.  ANd Debit cards are used the least for payments.  Credit card companies can add some offers/coupons for purchasing anything.  Customer should maintain a good credit score to avail best offers and loans for their needs. UPI merchants should also give some rewards or cash backs to promote UPI payments.  As UPI payments are Down Payments.  The volume of payments is increasing over the months from 2016 to 2018. |

|  |
| --- |
| 6.2)Count of orders based on the no. of payment installments |
| SELECT  payment\_installments,  COUNT(DISTINCT order\_id) AS count\_of\_orders FROM  `business-case-study-sql.Target\_dataset.payments` GROUP BY  payment\_installments |
|  |
| Sheets Plot: |
|  |
| Insights:  We observe that most of the payments are done at once. Then 2 to 10 instalments are used for payments.  And 11-24 instalments are very less used for payments. |