**Q1. Business Case: Target SQL**

**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 analysing 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 analysing the given dataset to extract valuable insights and provide actionable recommendations.

**What does 'good' look like?**

1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

   I have uploaded the Target dataset in Big Query; all the querying will be done using this.
   Upon initial review of The Target company's website, it appears to be a prominent e-commerce firm based in the USA, akin to platforms such as Amazon, Flipkart, Myntra etc, in India and offers an extensive array of products, encompassing categories like kids' items, back-to-school supplies, Halloween products, and more. The dataset in question seems to contain diverse fields related to orders, customer information, geolocation data, and other relevant aspects of the e-commerce operations.

   1. Data type of all columns in the "customers" table.

| Field name | Type |
|---|---|
| customer_id | STRING |
| customer_unique_id | STRING |
| customer_zip_code_prefix | INTEGER |
| customer_city | STRING |
| customer_state | STRING |

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

      The time range between which the orders were placed are 2016-09-04 to 2018-10-17.

```
select *
from `Target.orders`
order by order_purchase_timestamp
limit 10
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH

| Row | order_id | customer_id | order_status | order_purchase_timestamp | order_approved_at |
|---|---|---|---|---|---|
| 1 | 2e7a8482f6fb09756ca50c10d... | 08c5351a6aca1c1589a38f244... | shipped | 2016-09-04 21:15:19 UTC | 2016-10-07 13:18: |
| 2 | e5fa5a7210941f7d56d0208e4... | 683c54fc24d40ee9f8a6fc179f... | canceled | 2016-09-05 00:15:34 UTC | 2016-10-07 13:17: |
| 3 | 809a282bbd5dbcabb6f2f724fc... | 622e13439d6b5a0b486c4356... | canceled | 2016-09-13 15:24:19 UTC | 2016-10-07 13:16: |
| 4 | bfbd0f9bdef84302105ad712d... | 86dc2ffce2dfff336de2f386a78... | delivered | 2016-09-15 12:16:38 UTC | 2016-09-15 12:16: |
| 5 | 71303d7e93b399f5bcd537d12... | b106b360fe2ef8849fbbd056f7... | canceled | 2016-10-02 22:07:52 UTC | 2016-10-06 15:50: |
| 6 | 3b697a20d9e427646d925679... | 355077684019f7f60a031656b... | delivered | 2016-10-03 09:44:50 UTC | 2016-10-06 15:50: |
| 7 | be5bc2f0da14d8071e2d45451... | 7ec40b22510fdbea1b08921dd... | delivered | 2016-10-03 16:56:50 UTC | 2016-10-06 16:03: |
| 8 | 65d1e226dfaeb8cdc42f66542... | 70fc57eeae292675927697fe0... | canceled | 2016-10-03 21:01:41 UTC | 2016-10-04 10:18: |

**Insights:** In this data I have used the order_purchase_timestamp to check the time range in which the orders were placed. It includes all the order_status like 'delivered', 'canceled', 'invoiced', 'processing' etc.

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

The number of cities and states where customers have placed orders is 4119 and 27 respectively.

```
select count(distinct c.customer_city) City_count, count(distinct c.customer_state) State_count
from `Target.customers` as c left join `Target.orders` as o
on c.customer_id = o.customer_id
where o.order_id is not null and c.customer_id is not null
```

ery results

SAVE RESULTS

3 INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION

| City_count | State_count |
|---|---|
| 4119 | 27 |

2. **In-depth Exploration:**
   1. Is there a growing trend in the no. of orders placed over the past years?

In the below query I have extracted the year from order_purchase_timestamp and grouped it year wise and calculated the total number of orders.

```
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
COUNT(DISTINCT o.order_id) AS order_count
FROM `Target.orders` o JOIN `Target.customers` c
ON o.customer_id = c.customer_id
GROUP BY year
ORDER BY year
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DE

| Row | year | order_count |
|---|---|---|
| 1 | 2016 | 329 |
| 2 | 2017 | 45101 |
| 3 | 2018 | 54011 |

**Insights:** From the above data we can see an increasing trend in the number of customer orders over the given period.
**Recommendations**: It's important to note that the order count alone does not indicate the pace of business growth. To gain a more accurate understanding, we should also consider revenue growth also.

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

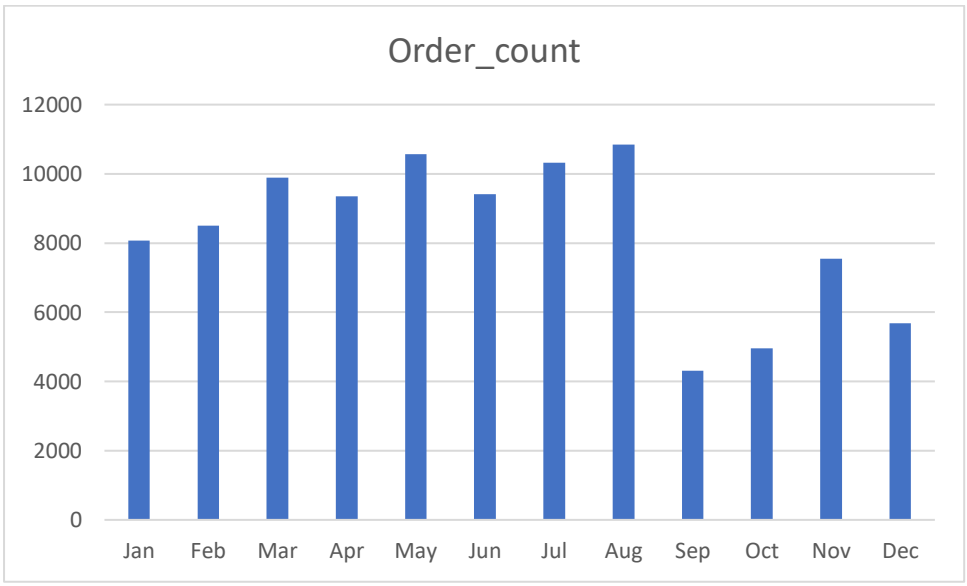
I have extracted the month from order_purchase_timestamp and grouped the whole dataset month wise and counted the order month wise.

```
--Q:Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
COUNT(DISTINCT order_id) AS order_count
FROM `Target.orders`
GROUP BY  month
ORDER BY  month
```

uery results  SAVE RESUL

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXEC

| | month | order_count |
|---|---|---|
| 1 | 1 | 8069 |
| 2 | 2 | 8508 |
| 3 | 3 | 9893 |
| 4 | 4 | 9343 |




**Insights:** I have tried to plot the data obtained from BigQuery in Excel. The count of orders generally increases from March to August with fluctuations in between. There is an increase in orders during February and March, coinciding with the Carnival season in Brazil.

**Recommendations:** It is important to note that further analysis with a larger dataset would be required to validate these seasonality trends. Also, we need to investigate the festival season and other special events that would take place in Brazil.

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
    - 0-6 hrs: Dawn
    - 7-12 hrs: Mornings
    - 13-18 hrs: Afternoon
    - 19-23 hrs: Night
    -

I have extracted the data for different time during the day and tried to group them.

```
SELECT CASE
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 7 AND 12 THEN 'Morning'
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
END AS hour,
COUNT(o.order_id) AS order_count
FROM `Target.orders` as o JOIN `Target.customers` c
ON o.customer_id = c.customer_id
GROUP BY  hour
ORDER BY  order_count DESC;
```

ery results SAVE RESULT

INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECU

| hour | order_count |
|---|---|
| Afternoon | 38135 |
| Night | 28331 |
| Morning | 27733 |
| Dawn | 5242 |

**Insights:** The analysis suggests that Brazilian customers show a preference for placing online orders during the daytime, particularly in the afternoon and night. This behaviour indicates that customers are more inclined to shop online when they have free time, potentially after work or other daily responsibilities.

**Recommendations:** To further improve sale, it can be recommended to provide offers, discounts, promotional campaigns etc, during the afternoon and nighttime while Brazilian's shop.

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

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

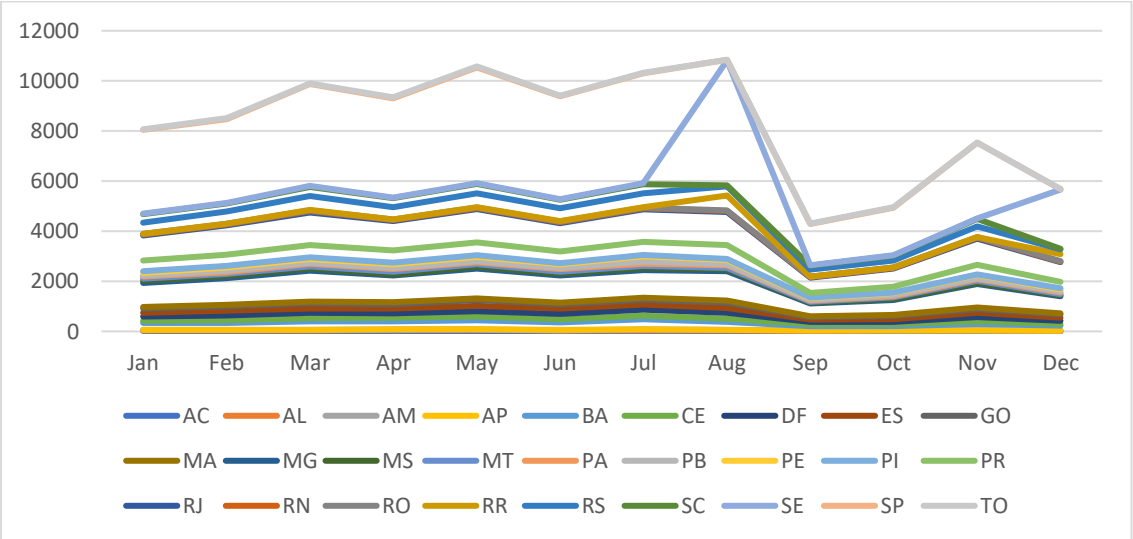
The order_purchase_timestamp is playing a major role in most of our queries. I have extracted the month from the above column and joined the orders table with customers table and then grouped it w.r.t to customer_state to get the month-on-month order of customers.

```sql
SELECT c.customer_state,
EXTRACT(month FROM o.order_purchase_timestamp) AS month,
COUNT(o.order_purchase_timestamp) AS order_count
FROM `Target.orders` o JOIN `Target.customers` c
ON o.customer_id = c.customer_id
GROUP BY  c.customer_state, month
ORDER BY  c.customer_state, month;
```

ery results

INFORMATION | RESULTS | JSON | EXECUTION DETAILS

| customer_state | month | order_count |
|---|---|---|
| AC | 1 | 8 |
| AC | 2 | 6 |
| AC | 3 | 4 |
| AC | 4 | 9 |
| AC | 5 | 10 |
| AC | 6 | 7 |




**Insights:** The depicted graph showcases the monthly order counts across various states in Brazil, offering valuable insights into state-specific customer purchasing patterns. São Paulo (SP) consistently leads with the highest order volumes each month, followed by Rio de Janeiro (RJ) and Minas Gerais (MG).

**Recommendations:** We should analyse the factors driving the substantial sales in São Paulo (SP), Rio de Janeiro (RJ), and Minas Gerais (MG) to identify strategies that could be replicated in other states to enhance sales. By understanding and implementing successful models from these leading states, we aim to boost sales across the board.

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

I have considered the customer_id column to group it w.r.t each state to get the customer distribution across the states.

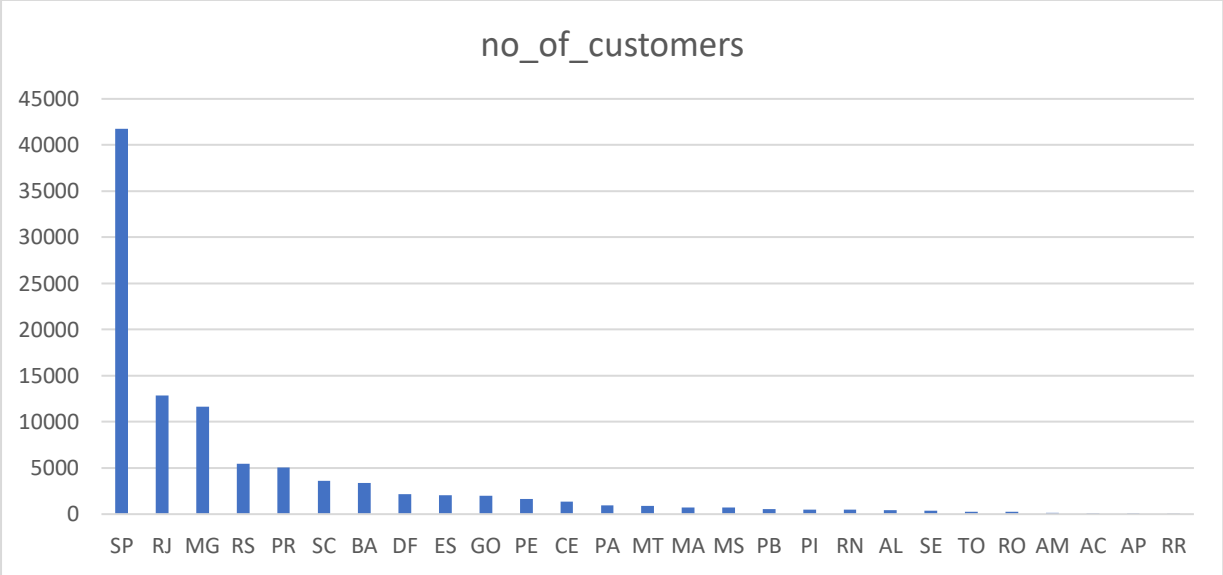
**Insights:** The data underscores that São Paulo (SP) boasts the largest customer base, largely attributed to its position as the most populous state in Brazil. This observation is consistent with our earlier analysis, highlighting a clear correlation between a state's population size and its order count.

```sql
SELECT c.customer_state,
COUNT(c.customer_id) AS no_of_customers
FROM `Target.customers` c
GROUP BY c.customer_state
ORDER BY no_of_customers DESC;
```

ery results

INFORMATION | RESULTS | JSON | EXECUTION DETAILS

| customer_state | no_of_customers |
|---|---|
| SP | 41746 |
| RJ | 12852 |
| MG | 11635 |
| RS | 5466 |
| PR | 5045 |
| SC | 3637 |




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

    1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only). You can use the "payment_value" column in the payments table to get the cost of orders.

    Formula used to calculate the % increase in the cost of orders from year 2017 to 2018 is (Sum of payment_value for 2018 - Sum of payment_value for 2017) divided by Sum of payment_value for 2017 for the months Jan to Aug.

```sql
SELECT EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
((SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 AND EXTRACT(MONTH FROM o.order_purchase_timestamp)
BETWEEN 1 AND 8 THEN p.payment_value END) -
SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM o.order_purchase_timestamp)
BETWEEN 1 AND 8 THEN p.payment_value END)) /
SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM o.order_purchase_timestamp)
BETWEEN 1 AND 8 THEN p.payment_value END))*100 AS percent_increase
FROM `Target.orders` o JOIN `Target.payments` p
ON o.order_id = p.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY 1
ORDER BY 1;
```

| month | percent_increase |
|---|---|
| 1 | 705.1266954171... |
| 2 | 239.9918145445... |
| 3 | 157.7786066709... |
| 4 | 177.8407701149... |
| 5 | 94.62734375677... |
| 6 | 100.2596912456... |
| 7 | 80.04245463390... |
| 8 | 51.60600520477... |

**Insights**: January shows the highest percentage increase, followed by February and April.

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

To gain insights into the price and freight values on a state level, we calculated the mean and sum of these values by a customer state.

```
SELECT c.customer_state,
ROUND(AVG(oi.price), 2) AS mean_price,
ROUND(SUM(oi.price), 2) AS total_price,
FROM `Target.orders` o JOIN `Target.order_items` oi
ON o.order_id = oi.order_id
JOIN `Target.customers` as c
ON o.customer_id = c.customer_id
group by c.customer_state
```

| customer_state | mean_price | total_price |
|---|---|---|
| MT | 148.3 | 156453.53 |
| MA | 145.2 | 119648.22 |
| AL | 180.89 | 80314.81 |
| SP | 109.65 | 5202955.05 |
| MG | 120.75 | 1585308.03 |
| PE | 145.51 | 262788.03 |
| RJ | 125.12 | 1824092.67 |
| DF | 125.77 | 302603.94 |
| RS | 120.34 | 750304.02 |
| SE | 153.04 | 58920.85 |

**Insights:** The analysis reveals interesting findings. While São Paulo (SP) has the highest total price value, it surprisingly has the lowest average/mean price value.

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

I have used the freight_value column in orders table to calculate the total and average for each state joined with order items table.

```sql
SELECT c.customer_state,
ROUND(AVG(oi.freight_value), 2) AS mean_freight_value,
ROUND(SUM(oi.freight_value), 2) AS total_freight_value
FROM `Target.orders` o JOIN `Target.order_items` oi
ON o.order_id = oi.order_id
JOIN `Target.customers` as c
ON o.customer_id = c.customer_id
group by c.customer_state
```

| customer_state | mean_freight_value | total_freight_value |
|---|---|---|
| MT | 28.17 | 29715.43 |
| MA | 38.26 | 31523.77 |
| AL | 35.84 | 15914.59 |
| SP | 15.15 | 718723.07 |
| MG | 20.63 | 270853.46 |
| PE | 32.92 | 59449.66 |
| RJ | 20.96 | 305589.31 |
| DF | 21.04 | 50625.5 |
| RS | 21.74 | 135522.74 |
| SE | 36.65 | 14111.47 |

**Insights:** Here also São Paulo (SP) has the highest total freight value and average freight value among all states.

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

1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
   Do this in a single query.
   You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
   - **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
   - **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

To ascertain the timeframe between placing an order, its delivery, and the projected delivery, we computed the days elapsed using the subsequent SQL query.

```sql
SELECT order_id, date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) as time_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY) as diff_estimated_delivery
FROM `Target.orders`
WHERE DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS NOT NULL
```

**Insights:** Most orders were delivered within the designated timeframe or on the expected delivery date. Only a small number of orders experienced delays in reaching customers, which warrants further investigation into the underlying causes.

Press Alt+

ery results

SAVE RESULTS | EXPL

INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH

| order_id | time_to_deliver | diff_estimated_delive |
|---|---|---|
| 1950d777989f6a877539f5379... | 30 | -12 |
| 2c45c33d2f9cb8ff8b1c86cc28... | 30 | 28 |
| 65d1e226dfaeb8cdc42f66542... | 35 | 16 |
| 635c894d068ac37e6e03dc54e... | 30 | 1 |
| 3b97562c3aee8bdedcb5c2e45... | 32 | 0 |
| 68f47f50f04c4cb6774570cfde... | 29 | 1 |

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

```
SELECT c.customer_state,
ROUND(AVG(oi.freight_value), 2) AS average_freight_value,
FROM `Target.orders` o JOIN `Target.order_items` oi
ON o.order_id = oi.order_id
JOIN `Target.customers` as c
ON o.customer_id = c.customer_id
group by c.customer_state
ORDER BY average_freight_value DESC
```

To obtain the top 5 highest and lowest average freight value, I have ordered the average_freight_value column in ascending and descending order respectively and tabulated in two different columns.

**Top 5 states with highest average freight value**

| Row | customer_state | average_freight_valu |
|---|---|---|
| 1 | RR | 42.98 |
| 2 | PB | 42.72 |
| 3 | RO | 41.07 |
| 4 | AC | 40.07 |
| 5 | PI | 39.15 |

**Top 5 states with lowest average freight value**

| | | |
|---|---|---|
| 23 | DF | 21.04 |
| 24 | RJ | 20.96 |
| 25 | MG | 20.63 |
| 26 | PR | 20.53 |
| 27 | SP | 15.15 |

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

```
SELECT c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)), 2) AS avg_delivery_time_Days
FROM `Target.orders` o JOIN `Target.order_items` oi
ON o.order_id = oi.order_id
JOIN `Target.customers` as c
ON o.customer_id = c.customer_id
group by c.customer_state
ORDER BY avg_delivery_time_Days DESC
```

To obtain the top 5 highest and lowest average delivery time, I have ordered the average_delivery_time_Days column in ascending and descending order respectively and tabulated in two different columns.

**Top 5 states with Highest Average Delivery time in DAYS**

| Row | customer_state | avg_delivery_time_Days |
|---|---|---|
| 1 | RR | 27.83 |
| 2 | AP | 27.75 |
| 3 | AM | 25.96 |
| 4 | AL | 23.99 |
| 5 | PA | 23.3 |

**Top 5 states with Lowest Average Delivery time in DAYS**

| Row | customer_state | avg_delivery_time_Days |
|---|---|---|
| 23 | SC | 14.52 |
| 24 | DF | 12.5 |
| 25 | MG | 11.52 |
| 26 | PR | 11.48 |
| 27 | SP | 8.26 |

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Query below:

```
SELECT  c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)), 2) AS delivery_really_fast
FROM `Target.orders` o JOIN `Target.order_items` i
ON o.order_id = i.order_id
JOIN `Target.customers` c ON o.customer_id = c.customer_id
GROUP BY  c.customer_state
ORDER BY delivery_really_fast desc
```

**Top 5 states where the delivery is really fast (In Days)**

| customer_state | delivery_really_fast |
|---|---|
| AC | 20.01 |
| RO | 19.08 |
| AM | 18.98 |
| AP | 17.44 |
| RR | 17.43 |

6. **Analysis based on the payments:**
    1. Find the month-on-month no. of orders placed using different payment types.

In order to grasp the patterns in payment methods, we examined the month-to-month order counts for various payment types by executing the provided SQL query.

```
SELECT p.payment_type,  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
COUNT(DISTINCT o.order_id) AS order_count
FROM `Target.orders` o JOIN `Target.payments` p
ON o.order_id = p.order_id
GROUP BY  p.payment_type, month
ORDER BY  p.payment_type, month
```

| payment_type | month | order_count |
|---|---|---|
| UPI | 1 | 1715 |
| UPI | 2 | 1723 |
| UPI | 3 | 1942 |
| UPI | 4 | 1783 |
| UPI | 5 | 2035 |

| Row | payment_type | order_count |
|---|---|---|
| 1 | UPI | 19784 |
| 2 | credit_card | 76505 |
| 3 | debit_card | 1528 |
| 4 | not_defined | 3 |

**Insights:** The analysis indicates a general upward trend observed between January and August, as well as another upward trend from September to November. Credit card transactions are the predominant choice for payments, closely followed by UPI. Conversely, debit card transactions are the least favoured payment option.

**Recommendations:** Since people are more preferring credit-card payment method, company can provide some discounts, cashbacks and other promotional offers for the customer who are using credit-card, to improve their sales further.

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

```
SELECT p.payment_installments, COUNT(o.order_id) AS order_count
FROM `Target.orders` o JOIN `Target.payments` p
ON o.order_id = p.order_id
WHERE o.order_status != 'canceled'
GROUP BY p.payment_installments
ORDER BY order_count DESC;
```

| payment_installment | order_count |
|---|---|
| 1 | 52184 |
| 2 | 12353 |
| 3 | 10392 |
| 4 | 7056 |
| 10 | 5292 |

**Insights:** The analysis shows that the most common scenario is a single payment instalment for the majority of orders.

## Actionable Insights & Recommendations

- The data indicates a notable disparity, with the state of São Paulo (SP) having considerably more orders than the cumulative total of the following five states. This highlights an opportunity for enhancement and growth in the other states.
- The Number of orders placed is highest during the Brazilian carnival i.e., during Feb to Mar.
- The data indicates a decrease in order numbers throughout September and October. Introducing discounts or promotional offers during these slower months could motivate customers to make purchases, potentially leading to an upswing in sales. Given that a significant portion of customers favor credit card payments, extending discounts, cashbacks, and special promotions to credit card users might further enhance sales.
- Although economic condition data is not part of the current dataset, conducting an analysis to assess its influence on sales can provide valuable insights.