






# Target SQL Project

Name: Soham Suryawanshi

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

**Soln:** In BigQuery, looking at the schema of the 'customers' table we can see the data types of columns.


	customers	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT
SCHEMA   DETAILS   PREVIEW   LINEAGE					
Filter Enter property name or value					
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default value
<input type="checkbox"/>	<a href="#">customer_id</a>	STRING	NULLABLE		
<input type="checkbox"/>	<a href="#">customer_unique_id</a>	STRING	NULLABLE		
<input type="checkbox"/>	<a href="#">customer_zip_code_prefix</a>	INTEGER	NULLABLE		
<input type="checkbox"/>	<a href="#">customer_city</a>	STRING	NULLABLE		
<input type="checkbox"/>	<a href="#">customer_state</a>	STRING	NULLABLE		

## 2. Time period for which the data is given

**Soln:** To get the Time period, we have to choose an "orders" dataset and apply the below query.

```
SELECT MIN(order_purchase_timestamp)AS  
First_purchase_date,MAX(order_delivered_customer_date)AS  
Last_delivery_date  
FROM `sql_case_study_target.orders`
```

**Output:**

Query results 

JOB INFORMATION		RESULTS	JSON	EXECUTION DET
Row	First_purchase_date	Last_delivery_date		
1	2016-09-04 21:15:19 UTC	2018-10-17 13:22:46 UTC		

**Conclusion:** Data is given from **4th Sept 2016** to **17th Oct 2018**

### 3. Cities and States of customers ordered during the given period

**Soln:** Since we already know that data we have is from 2016 to 2018 so we can directly select the “**customers**” dataset to perform the analysis. Below we can see the query used to get the conclusion for customers where the order was delivered.

```
SELECT DISTINCT customer_city , customer_state
FROM `sql_case_study_target.customers` AS c
INNER JOIN
`sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
WHERE o.order_status = "delivered"
```

**Output:** Showing result for few customers for an example.

### Q2) 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?

**Soln:** Below is the query which helps us gain some insights on whether the trend was growing or not.

```

SELECT EXTRACT(Year FROM order_purchase_timestamp) AS Year,

EXTRACT(Month FROM order_purchase_timestamp) AS Month,

COUNT(*) AS Total_orders

FROM `sql_case_study_target.orders`

GROUP BY Year,Month

ORDER BY Year,Month

```

### Result:

Query results				
JOB INFORMATION		RESULTS	JSON	
Row	Year	Month	Total_orders	
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	

### Insights:

- We can say that there was a growing trend on e-commerce in Brazil.
- Based on the results, we can also conclude that the **highest number** of orders happened during the months which are either **close to the year end** or are **near to the new year**.
- However, Overall trend has been **growing since 2016 to 2018** if we compare the average number of orders year wise.

### Recommendation:

- If we consider the flow of orders, there were some of the months where the number of orders were lesser compared to other months. Might be possible that those were not seasonal months but still it could have been better if some offers/discounts were provided on the products specifically targeted in those months.

## 2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

### Soln:

```
SELECT SUM(CASE WHEN Hour BETWEEN 0 AND 6 THEN Counts ELSE 0 END) AS Dawn,  
SUM(CASE WHEN Hour BETWEEN 7 AND 12 THEN Counts ELSE 0 END) AS Morning,  
SUM(CASE WHEN Hour BETWEEN 13 AND 18 THEN Counts ELSE 0 END) AS Afternoon,  
SUM(CASE WHEN Hour BETWEEN 19 AND 23 THEN Counts ELSE 0 END) AS Night  
FROM (SELECT EXTRACT(HOUR FROM order_purchase_timestamp) AS Hour,  
COUNT(DISTINCT order_id) AS Counts  
FROM `sql_case_study_target.orders`  
WHERE order_status = "delivered"  
GROUP BY Hour ORDER BY Hour)
```

### Result:

### Query Output:

Row	Dawn	Morning	Afternoon	Night
1	5072	26919	36965	27522

### Insights:

As we can see from the output, most of the purchases happen in the **Afternoon** followed by **Night** followed by **Morning** and then **Dawn**.

So Brazilian customers tend to buy more in the **Afternoon**.

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

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

Soln:

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

This query will give us orders for every month for every state on a yearly basis.  
So it will be easy to analyze in which month how many orders were placed.

Query Results:

Row	Year	Month	Orders_Count	customer_state
1	2016	9	1	RR
2	2016	9	1	RS
3	2016	9	2	SP
4	2016	10	113	SP
5	2016	10	24	RS
6	2016	10	4	BA

We can see month on month orders placed for every state on a yearly basis.

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

Soln:

```
SELECT COUNT(DISTINCT customer_id) AS total_customers, customer_state
FROM `sql_case_study_target.customers`
GROUP BY customer_state
ORDER BY total_customers DESC
```

Query Results:

Row	total_customers	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF

By running the above query we can see customer distribution for every state. As we can see, the maximum customer base is in **Sao Paulo State** in **Brazil**.

## Q 4. Impact on Economy: Analyze 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.

Soln:

```

WITH cte AS

(SELECT SUM(payment_value)AS revenue,EXTRACT(YEAR FROM
order_purchase_timestamp) AS Year

FROM `sql_case_study_target.orders` AS o

INNER JOIN `sql_case_study_target.payments` AS p

ON o.order_id = p.order_id

WHERE EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 0 AND 8

GROUP BY Year),

both_revenue AS

(SELECT cte.revenue,LEAD(cte.revenue,1) OVER(ORDER BY cte.Year DESC) AS
last_revenue, cte.YEAR FROM cte)

SELECT both_revenue. Year, both_revenue.revenue, both_revenue.last_revenue,
ROUND(((both_revenue.revenue -
both_revenue.last_revenue)/both_revenue.last_revenue) * 100,2) AS
percent_increase

FROM both_revenue

```

We have used the above query to find out percent increase in cost of orders from **2017 to 2018** for months between **Jan and Aug**.

### Query Result:

Row	Year	revenue	last_revenue	percent_increase
1	2018	8694733.839999...	3669022.119999...	136.98
2	2017	3669022.119999...	null	null

### Insights:

As we can see, the **percent increase from Year 2017 to Year 2018 is almost 136%**. By this we can say that the company took a large growth jump of more than 100% of what it did last year in those specified months.

So it is advisable to invest more in those months to develop some marketing strategies so that more revenue can be expected in the subsequent years as well.

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

Soln:

```
SELECT SUM(oi.price) AS Total_Val, AVG(oi.price) AS Avg_Val, c.customer_state
FROM `sql_case_study_target.customers` AS c
INNER JOIN `sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
INNER JOIN `sql_case_study_target.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
```

Query Results:

Row	Total_Val	Avg_Val	customer_state
1	156453.5299999...	148.2971848341...	MT
2	119648.2199999...	145.2041504854...	MA
3	80314.81	180.8892117117...	AL
4	5202955.050001...	109.6536291597...	SP
5	1585308.029999...	120.7485741488...	MG
6	262788.0299999...	145.5083222591...	PE
7	1824092.669999...	125.1178180945...	RJ
8	302603.9399999...	125.7705486284...	DF

As we can see here the query gives results for **Average** and **Total** value for each state.

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

Soln:

```
SELECT SUM(oi.freight_value) AS Total_Val, AVG(oi.freight_value) AS Avg_Val
, c.customer_state
FROM `sql_case_study_target.customers` AS c
INNER JOIN `sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
INNER JOIN `sql_case_study_target.order_items` AS oi
```



```
ON o.order_id = oi.order_id
GROUP BY c.customer_state
```

#### Query Output:

Row	Total_Val	Avg_Val	customer_state
1	18860.09999999...	35.65236294896...	RN
2	48351.58999999...	32.71420162381...	CE
3	135522.7400000...	21.73580433039...	RS
4	89660.26000000...	21.47036877394...	SC
5	718723.0699999...	15.14727539041...	SP
6	270853.4600000...	20.63016680630...	MG
7	100156.6799999...	26.36395893656...	BA

Above query gives **Total** and **Average** value of Order freight for each **state**.

### Q5. Analysis based on sales, freight and delivery time.

- 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.

Soln:

```
SELECT
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)AS
time_to_deliver,
DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY) AS
diff_estimated_delivery
FROM `sql_case_study_target.orders`
```

`ORDER BY time_to_deliver DESC,diff_estimated_delivery`

#### Query Output:

Row	time_to_deliver	diff_estimated_delivery
1	209	181
2	208	188
3	195	165
4	194	155
5	194	161
6	194	166
7	191	175
8	189	167

#### Analysis & Observation:

For the first record, the order **should have been delivered in 181 days**, it was **delivered in 209 days**, a **delay of 28 days**.

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

Soln:

a) Top 5 States with highest average freight value:

```
SELECT c.customer_state, AVG(freight_value) AS avg_freight_value
FROM `sql_case_study_target.customers` AS c
INNER JOIN `sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
INNER JOIN `sql_case_study_target.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY avg_freight_value DESC
LIMIT 5
```

Query Output:

Row	customer_state	avg_freight_value
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

b) Top 5 States with lowest average freight value:

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

Query Output:

Row	customer_state	avg_freight_value
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

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

Soln:

a) Top 5 States with Highest Average Delivery Time

```
SELECT c.customer_state,
AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
avg_delivery_time
```

```

FROM `sql_case_study_target.customers` AS c
INNER JOIN `sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_time DESC
LIMIT 5

```

Query Output:

Row	customer_state	avg_delivery_time
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...

#### b) Top 5 States with Lowest Average Delivery Time

```

SELECT c.customer_state,
AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, DAY)) AS
avg_delivery_time
FROM `sql_case_study_target.customers` AS c
INNER JOIN `sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_time ASC
LIMIT 5

```

Query Output:

Row	customer_state	avg_delivery_time
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...
4	DF	12.50913461538...
5	SC	14.47956019171...

5. 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.

**Soln:**

```
SELECT c.customer_state,
AVG(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date, DAY)) AS
avg_diff_estimate_and_real
FROM `sql_case_study_target.customers` AS c
INNER JOIN `sql_case_study_target.orders` AS o
ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_diff_estimate_and_real ASC
```

**Query Output:**

Row	customer_state	avg_diff_estimate_and_real
1	AC	-19.7625
2	RO	-19.131687242798357
3	AP	-18.731343283582081
4	AM	-18.606896551724137
5	RR	-16.414634146341459

**Insights:**

More negative avg\_diff\_estimated\_delivery means the order delivered really very fast compared to the estimated date

**Q6.Analysis based on the payments:**

1. Find the month on month no. of orders placed using different payment types.

**Soln:**

```
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS YEAR,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS Month,
COUNT(o.order_id) AS no_of_orders,
p.payment_type AS payment_type
FROM `sql_case_study_target.payments` AS p
INNER JOIN `sql_case_study_target.orders` AS o
ON p.order_id = o.order_id
GROUP BY Year,Month,payment_type ORDER BY Year,Month
```

**Query Output:**

Row	YEAR ▼	Month ▼	no_of_orders ▼	payment_type ▼
1	2016	9	3	credit_card
2	2016	10	254	credit_card
3	2016	10	23	voucher
4	2016	10	2	debit_card
5	2016	10	63	UPI
6	2016	12	1	credit_card
7	2017	1	61	voucher
8	2017	1	197	UPI

### Insights:

As we can see, here is the distribution of month on month number of orders with payment being received by different payment types including credit card, voucher, debit card, UPI. In 2016 mostly we can see credit card payments compared to any other payment type.

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

Soln:

```
SELECT p.payment_installments, COUNT(o.order_id) AS no_of_orders
FROM `sql_case_study_target.orders` AS o
INNER JOIN `sql_case_study_target.payments` AS p
ON o.order_id = p.order_id
GROUP BY p.payment_installments
ORDER BY no_of_orders DESC
```

Query Output:

Row	payment_installments	no_of_orders ▼
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920

### Insights:

This data provides insights into the patterns of increased orders based on the number of installments, allowing for a comparison of total order values for better recovery strategies. The information will aid management and the recovery team in mitigating risks effectively.

### Recommendations:

1. There is a noticeable **increase in sales** during **March 2017/18**, as well as similar patterns in **April and August 2017/18**. However, apart from these periods, no clear seasonal or peak season patterns emerge. Therefore, it is recommended that the **business focuses on stocking up inventory** and offering **active discounts** during seasonal times to **maximize the effectiveness** of orders.
2. Based on the analysis, **customers** are more **active** during the **afternoon and night**. To capitalize on this trend, it is recommended that the business team provides **additional discounts** and **reduces delivery charges** during these time periods, potentially leading to an increase in orders.
3. States such as **Minas, Rio de Janeiro, and Sao Paulo** exhibit **higher order counts** compared to other states, making them hotspots for our company. To ensure smooth operations for customers, it is suggested to dedicate a team exclusively to these states, facilitating efficient procedures and enhancing customer satisfaction.
4. After analyzing the freight charges and delivery, it is advisable for the business to engage in discussions with delivery partners to **improve their service quality**. **Replacing slow delivery teams** with providers offering **better service** can help reduce

additional costs and make a significant difference in terms of customer satisfaction, which is crucial for the success of the business.

5. Payment analysis reveals that a **majority of customers** prefer using **credit cards** for their transactions. Therefore, it is recommended that management collaborates with more private banks to offer attractive deals and **discounts on credit cards** and **UPI payments**. Such offers can attract customers and potentially lead to an **increase in sales**.