Business Case Study – Target SQL

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

Initial exploration of dataset

1. Data type of columns in a table - String,Interger,float,timestamp

2. Time period for which the data is given - Date is from 2016 TO 2018

3. Cities and States of customers ordered during the given period - Customer data is from 27 States and 50 Cities

Analysis of tables in the dataset

Total 8 tables are there in the dataset

Customer Table – Customer data is from 27 States and 50 Cities

Primary Keys – customer_id

Foreign Keys - customer_unique_id,customer_zip_code_prefix

Geolocation Table – We can get the full name of the cities from this table using zipcode as key

Primary Key - geolocation_zip_code_prefix

Order_items – freight_value and price for the order is in this table

PK – order_id,product_id,seller_id

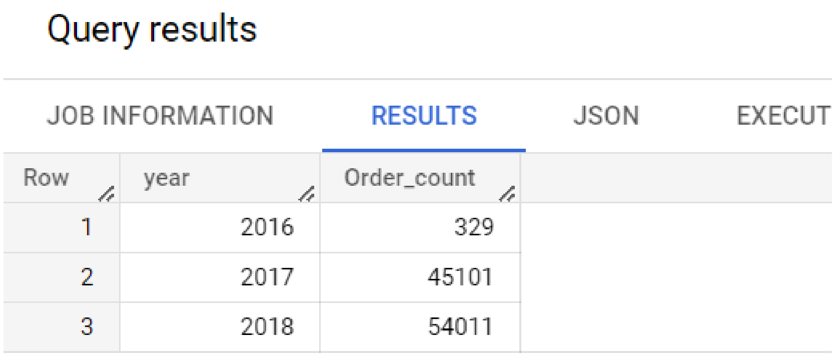
Note :Freight value is the price paid or payable to the exporter for the cargo when it is unloaded from the shipper at the port when imported

Order_reviews – review_comment_title contains the review and where no review is given it is null

Que: 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?

```
SELECT year,Order_count
FROM (SELECT COUNT(order_id)AS Order_Count, EXTRACT(Year FROM o.orde
r_purchase_timestamp ) AS year
FROM `scaler-dsml-sql-380819.Target_SQL_Project.orders` AS o
GROUP BY EXTRACT(Year FROM o.order_purchase_timestamp )) AS x
ORDER BY year
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUT

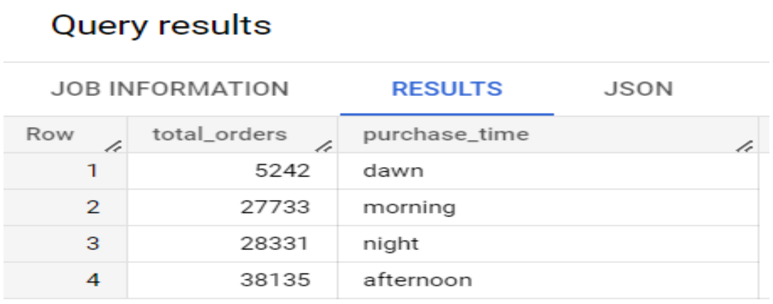
Row	year	Order_count
1	2016	329
2	2017	45101
3	2018	54011

Insights - The YoY trend is positive from 2016 to 2018.

Recommendation - Introducing seasonal/new inventory to lower the impact of the decrease in sales.

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

```
SELECT count(orders)AS total_orders,
CASE
WHEN hour BETWEEN 0 AND 6 THEN 'dawn'
WHEN hour BETWEEN 6 AND 12 THEN 'morning'
WHEN hour BETWEEN 12 AND 18 THEN 'afternoon'
WHEN hour BETWEEN 18 AND 23 THEN 'night'
END AS purchase_time
FROM(SELECT o.order_id AS orders,EXTRACT(HOUR FROM o.order_purchase_
timestamp) AS hour
FROM `scaler-dsml-sql-380819.Target_SQL_Project.orders` AS o)
GROUP BY purchase_time
ORDER BY total_orders
```



Query results

JOB INFORMATION | RESULTS | JSON

Row	total_orders	purchase_time
1	5242	dawn
2	27733	morning
3	28331	night
4	38135	afternoon

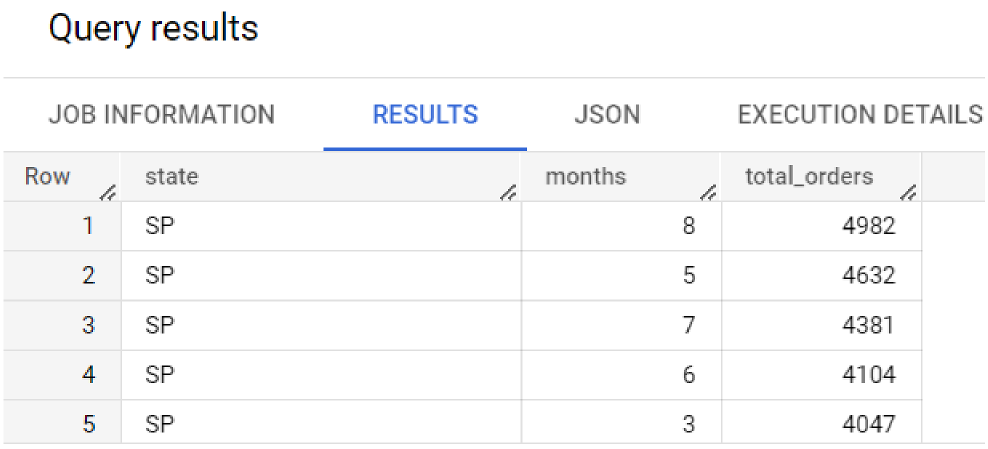
Insights – Maximum orders are in Afternoon and Minimum orders are in Dawn.

Recommendation - Offering discounts for dawn timings to increase sales.

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

1.Get month on month orders by states

```
SELECT state,months,COUNT(orders) AS total_orders
FROM(
SELECT c.customer_state AS state,EXTRACT(MONTH FROM o.order_purchase
_timestamp)AS months,o.order_id AS orders
FROM `scaler-dsml-sql-380819.Target_SQL_Project.orders` AS o
JOIN  `scaler-dsml-sql-
380819.Target_SQL_Project.customers` AS c ON o.customer_id=c.custome
r_id)
GROUP BY state,months
ORDER BY total_orders DESC
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

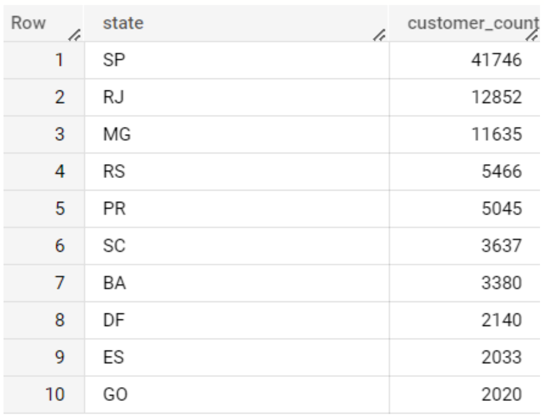
Row	state	months	total_orders
1	SP	8	4982
2	SP	5	4632
3	SP	7	4381
4	SP	6	4104
5	SP	3	4047

Insights – SP has the most numbers of orders

Recommendation- State with highest orders keep the warehouses in these state stocked to avoid shortage and in delay of delivery of product

2.Distribution of customers across the states in Brazil

```
SELECT c.customer_state AS state,COUNT(c.customer_id) AS customer_co
unt
FROM `scaler-dsml-sql-380819.Target_SQL_Project.customers` AS c
GROUP BY state
ORDER BY customer_count DESC
LIMIT 10
```



Row	state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Insights – Above states are top 10 states with highest customers

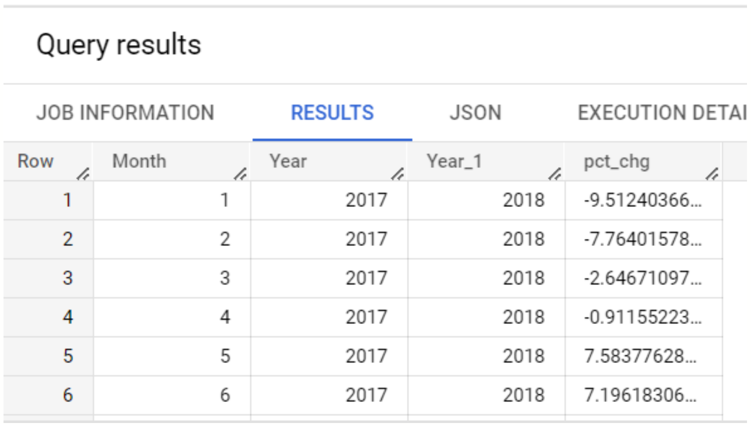
Que: 4Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

```
 WITH payment_analysis AS (

SELECT  EXTRACT(Year FROM o.order_purchase_timestamp ) AS Year,EXTRA
CT(Month FROM o.order_purchase_timestamp )AS Month,p.payment_value
FROM `scaler-dsml-sql-380819.Target_SQL_Project.orders` AS o
JOIN `scaler-dsml-sql-
380819.Target_SQL_Project.payments` AS p ON o.order_id=p.order_id),
payment_sum_analysis AS (SELECT Month,Year,avg(payment_value) AS sum
_payments FROM payment_analysis
GROUP BY Year,Month
HAVING Month<=8 AND Year BETWEEN 2017 AND 2018)
SELECT a.Month,a.Year,b.Year,((b.sum_payments-
a.sum_payments)/a.sum_payments)*100 AS pct_chg
FROM payment_sum_analysis AS a
JOIN payment_sum_analysis AS b ON a.Month=b.Month AND a.Year!=b.Year

ORDER BY a.Year,a.Month
LIMIT 8
```

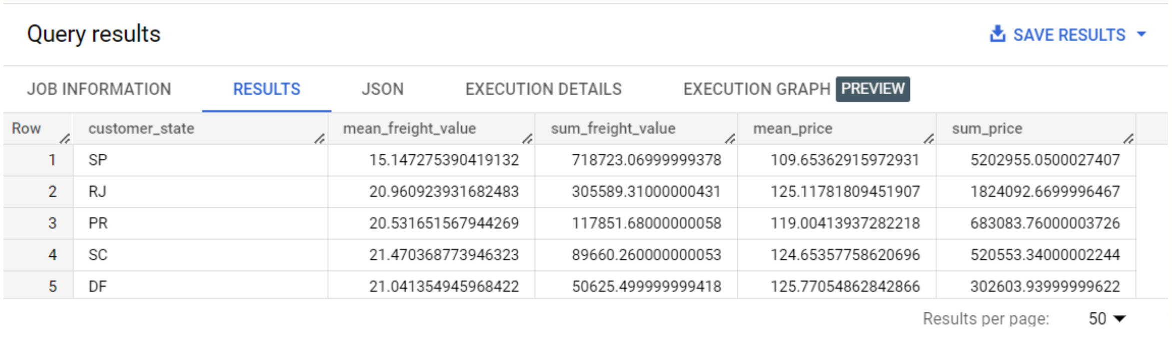

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAI

Row	Month	Year	Year_1	pct_chg
1	1	2017	2018	-9.51240366...
2	2	2017	2018	-7.76401578...
3	3	2017	2018	-2.64671097...
4	4	2017	2018	-0.91155223...
5	5	2017	2018	7.58377628...
6	6	2017	2018	7.19618306...

2.Mean & Sum of price and freight value by customer state

```
SELECT c.customer_state,AVG(ot.freight_value) AS mean_freight_value,
SUM(ot.freight_value)AS sum_freight_value,AVG(ot.price)mean_price,SU
M(ot.price) AS sum_price
FROM `scaler-dsml-sql-380819.Target_SQL_Project.order_items` AS ot
JOIN `scaler-dsml-sql-
380819.Target_SQL_Project.orders`AS o ON o.order_id=ot.order_id
JOIN `scaler-dsml-sql-
380819.Target_SQL_Project.customers`AS c ON c.customer_id=o.customer
_id
GROUP BY c.customer_state
```


Query results

SAVE RESULTS

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

Row	customer_state	mean_freight_value	sum_freight_value	mean_price	sum_price
1	SP	15.147275390419132	718723.06999999378	109.65362915972931	5202955.0500027407
2	RJ	20.960923931682483	305589.31000000431	125.11781809451907	1824092.6699996467
3	PR	20.531651567944269	117851.68000000058	119.00413937282218	683083.76000003726
4	SC	21.470368773946323	89660.260000000053	124.65357758620696	520553.34000002244
5	DF	21.041354945968422	50625.499999999418	125.77054862842866	302603.93999999622

Results per page: 50

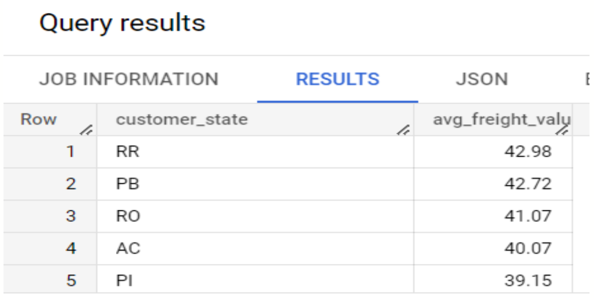
Que: 5 Analysis on sales, freight and delivery time

Sort the data to get the following:

- Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
SELECT c.customer_state,ROUND((AVG(ot.freight_value)),2) AS avg_frei
ght_value
FROM `scaler-dsml-sql-380819.Target_SQL_Project.order_items` AS ot
JOIN `scaler-dsml-sql-
380819.Target_SQL_Project.orders`AS o ON o.order_id=ot.order_id
JOIN `scaler-dsml-sql-
380819.Target_SQL_Project.customers`AS c ON c.customer_id=o.customer
_id
GROUP BY c.customer_state
```

```
ORDER BY avg_freight_value DESC
LIMIT 5
```



Query results

JOB INFORMATION | RESULTS | JSON

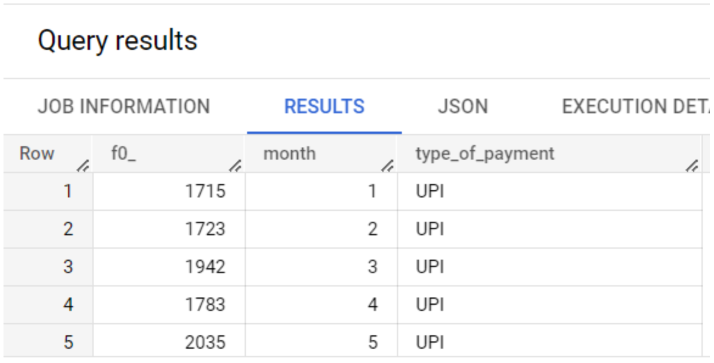
Row	customer_state	avg_freight_valu
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

Que: 6 Payment type analysis:

1. Month over Month count of orders for different payment types

```
SELECT COUNT(orders),month,type_of_paymentFROM(SELECT o.order_id ord
ers,EXTRACT (MONTH FROM o.order_purchase_timestamp) AS month,p.payme
nt_type AS type_of_payment

FROM `scaler-dsml-sql-380819.Target_SQL_Project.orders` AS o
JOIN `scaler-dsml-sql-
380819.Target_SQL_Project.payments` AS p ON o.order_id=p.order_id)
GROUP BY month,type_of_payment
ORDER BY type_of_payment,month
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET.

Row	f0_	month	type_of_payment
1	1715	1	UPI
2	1723	2	UPI
3	1942	3	UPI
4	1783	4	UPI
5	2035	5	UPI

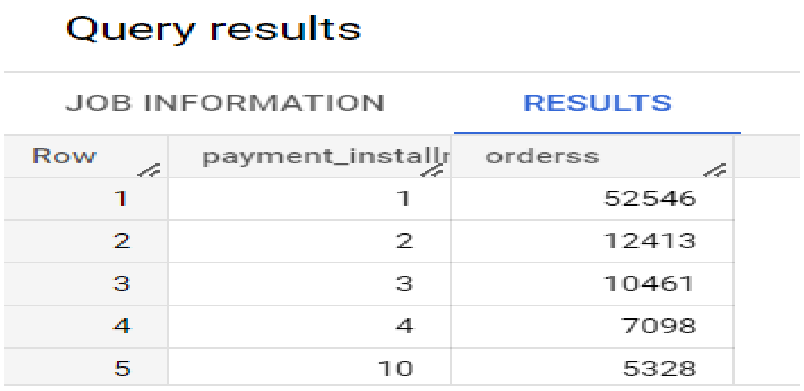
Insight - The lowest amount of transactions is done through UPI.

Recommendation – Provide discounts on UPI transactions to increase sales where people use UPI .As Not every person has a credit card but UPI transactions are also increasing we can use this to increase more sales.

2.Count of orders based on the no. of payment installments

```
SELECT payment_installments,COUNT(order_id) AS orderss
FROM `scaler-dsml-sql-380819.Target_SQL_Project.payments`
```

```
GROUP BY payment_installments
ORDER BY orderss DESC
```



Row	payment_installr	orderss
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328

Insights – People make payments in less installments above are top5 number_of_installements in which people pay.

Recommendation- Provide them No cost EMI options for short installments.