

PART 1

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

Q1: Datatype of all the columns in tables?

Query: `SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE from Ecommerce.INFORMATION_SCHEMA.COLUMNS where table_name in ('customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', 'sellers') order by TABLE_NAME;`

Result:

Row	TABLE_NAME	COLUMN_NAME	DATA_TYPE
1	customers	customer_id	STRING
2	customers	customer_state	STRING
3	customers	customer_zip_code_prefix	INT64
4	customers	customer_unique_id	STRING
5	customers	customer_city	STRING
6	geolocation	geolocation_city	STRING
7	geolocation	geolocation_lat	FLOAT64
8	geolocation	geolocation_state	STRING
9	geolocation	geolocation_lng	FLOAT64
10	geolocation	geolocation_zip_code_prefix	INT64

In the above result, we can see the datatypes of all columns for each table.

Q2: Time period for which the data is given?

Query: `select distinct (extract(year from order_purchase_timestamp)) as order_years from Ecommerce.orders;`

Result:

Row	order_years
1	2017
2	2018
3	2016

In the above result, we can see the time period for the data given is from year 2016 to 2018. We can find the months also for which the data is given by following query.

Query: `SELECT`

```
MAX((EXTRACT( month FROM order_purchase_timestamp))) as max_month, MIN((EXTRACT( month FROM
order_purchase_timestamp))) as min_month,
(EXTRACT( year FROM order_purchase_timestamp)) as order_years
FROM Ecommerce.orders GROUP BY EXTRACT( year FROM order_purchase_timestamp);
```

Result:

Row	max_month	min_month	order_years
1	12	1	2017
2	10	1	2018
3	12	9	2016

In the above table, we can see that the data given is from 09/2016 (Sept, 2016) to 10/2018 (Oct, 2018).

Q3: Cities and states covered?

Query for cities:

```
select distinct customer_city as cities_covered from Ecommerce.customers
order by customer_city;
```

Result:

Row	cities_covered
1	abadia dos dourados
2	abadiania
3	abaete
4	abaetetuba
5	abaiara
6	abaira
7	abare
8	abatia
9	abdon batista
10	abelardo luz

From the above query, we find the cities covered by Target in Brazil. The number of cities that we get from this is 4119.

Query for states: `select distinct customer_state as states_covered from Ecommerce.customers order by customer_state;`

Result:

Row	states_covered
1	AC
2	AL
3	AM
4	AP
5	BA
6	CE
7	DF
8	ES
9	GO
10	MA

In the above table, we can see the states covered by Target in Brazil. The number of states that we get from this query is 27.

Query for no. of cities and states covered-

`select count(distinct customer_city) as cities_covered, count(distinct customer_state) as states_covered from `Ecommerce.customers`;`

Result-

Row	cities_covered	states_covered
1	4119	27

Part2

In-depth Exploration

Q1- 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?

Query for Years:

```
select order_year, count(order_id) as no_of_orders from (select extract(year from order_purchase_timestamp) as order_year, order_id from `Ecommerce.orders`) group by order_year order by order_year;
```

Result-

Row	order_year	no_of_orders
1	2016	329
2	2017	45101
3	2018	54011

From the above table, we can see the no. of orders are increasing every year so we can say that there is an increasing trend on e-commerce in Brazil.

Query for months:

```
select order_month, count(order_id) as no_of_orders from (select extract(month from order_purchase_timestamp) as order_month, order_id from `Ecommerce.orders`) group by order_month order by no_of_orders desc;
```

Result:

Row	order_month	no_of_orders
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674
11	10	4959
12	9	4305

In the above result, we can see that the maximum orders are placed in 8th Month (August) and then 2nd highest in 5th Month (May).

Query for Days:

```
select order_day, count(order_id) as no_of_orders from (select format_datetime('%A',order_purchase_timestamp) as order_day, order_id from `Ecommerce.orders`) group by order_day order by no_of_orders desc;
```

Result:

Row	order_day	no_of_orders
1	Monday	16196
2	Tuesday	15963
3	Wednesday	15552
4	Thursday	14761
5	Friday	14122
6	Sunday	11960
7	Saturday	10887

Q2: What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query: select

```
case
when order_hour between 4 and 6 then 'Dawn'
when order_hour between 6 and 12 then 'Morning'
when order_hour between 12 and 18 then 'Afternoon'
else 'Night' end as order_time_period, count(order_id) as no_of_orders from (select extract
(hour from order_purchase_timestamp) as order_hour, order_id from `Ecommerce.orders`)
group by order_time_period order by no_of_orders desc;
```

Result:

Row	order_time_period	no_of_orders
1	Afternoon	38135
2	Night	32677
3	Morning	27733
4	Dawn	896

We can see that Brazilian customers tend to buy in the afternoon that is from 12PM to 6PM.

Part 3

Evolution of E-commerce orders in the Brazil region

Q1: Get month on month orders by region, states.

Query for month on month city-wise orders:

```
select customer_city, order_month, count(order_id) as no_of_orders from (select extract(month from order_purchase_timestamp) as order_month, order_id, customer_city from `Ecommerce.orders` ord join `Ecommerce.customers` cus on ord.customer_id = cus.customer_id) group by customer_city, order_month order by customer_city, order_month;
```

Result:

Row	customer_city	order_month	no_of_orders
1	abadia dos dourados	3	1
2	abadia dos dourados	7	1
3	abadia dos dourados	9	1
4	abadiania	1	1
5	abaete	2	1
6	abaete	3	2
7	abaete	5	1
8	abaete	6	2
9	abaete	7	2
10	abaete	8	2

Query for month on month State-wise orders:

```
select customer_state, order_month, count(order_id) as no_of_orders from (select extract(month from order_purchase_timestamp) as order_month, order_id, customer_state from `Ecommerce.orders` ord join `Ecommerce.customers` cus on ord.customer_id = cus.customer_id) group by customer_state, order_month order by customer_state, order_month;
```

Result:

Row	customer_state	order_month	no_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Query for city-wise orders:

```
select customer_city, count(order_id) as no_of_orders from `Ecommerce.orders` ord join  
`Ecommerce.customers` cus on ord.customer_id = cus.customer_id group by customer_city order  
by count(order_id) desc;
```

Result:

Row	customer_city	no_of_orders
1	sao paulo	15540
2	rio de janeiro	6882
3	belo horizonte	2773
4	brasilia	2131
5	curitiba	1521
6	campinas	1444
7	porto alegre	1379
8	salvador	1245
9	guarulhos	1189
10	sao bernardo do campo	938

In the above table, we can see that highest no of orders are placed in City Sao Paulo which is the financial centre of Brazil and Rio De Janeiro is 2nd which is the capital of Brazil.

Query for state-wise orders:

```
select customer_state, count(order_id) as no_of_orders from `Ecommerce.orders` ord join  
`Ecommerce.customers` cus on ord.customer_id = cus.customer_id group by customer_state  
order by count(order_id) desc;
```

Result:

Row	customer_state	no_of_orders
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

Highest no. of orders is placed in Sao Paulo (SP) state followed by Rio De Janeiro (RJ) state.

Q2- How are customers distributed in Brazil?

Query:

```
select customer_state, count(distinct customer_id) as Active_customers_per_state,  
count(distinct customer_unique_id) as Reg_customers_per_state from `Ecommerce.customers`  
group by customer_state order by count(distinct customer_id) desc;
```

Result:

Row	customer_state	Active_customers_per_state	Reg_customers_per_state
1	SP	41746	40302
2	RJ	12852	12384
3	MG	11635	11259
4	RS	5466	5277
5	PR	5045	4882
6	SC	3637	3534
7	BA	3380	3277
8	DF	2140	2075
9	ES	2033	1964
10	GO	2020	1952

It can be inferred from the above result that Sao Paulo (SP) state has the highest no of active and registered customers followed by Rio De Janeiro (RJ) and Minas Gerais (MG).

Part 4

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

Q1: Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only).

Query for year on year:

```
with cost_of_orders as (select sum(cost_of_order) as cost_per_year, year_of_order from
(select price + freight_value as cost_of_order, EXTRACT(month from shipping_limit_date) as
month_of_order, Extract(year from shipping_limit_date) as year_of_order from
`Ecommerce.order_items`) where month_of_order between 1 and 8 group by year_of_order)

select ((lead(cost_per_year) over(order by year_of_order)-
cost_per_year)/cost_per_year)*100 as percentage_increase_in_orders_cost from
cost_of_orders limit 1;
```

Result-

Row	percentage_increase_in_orders_cost
1	153.79

We can see the % increase in cost of orders from 2017 to 2018 is 153.79%. To get the % increase month on month, we can use the following query.

Query for month on month:

```
WITH cost_of_orders_2017 as (SELECT month_of_order, sum(cost) as cost_2017 from (SELECT Ext
ract(year from shipping_limit_date) as year_2017, EXTRACT(month from shipping_limit_date) a
s month_of_order, price + freight_value as cost from `Ecommerce.order_items`
where Extract(year from shipping_limit_date) = 2017 )
group by month_of_order order by month_of_order),
```

```
cost_of_orders_2018 as (SELECT month_of_order, sum(cost) as cost_2018 from (SELECT Extract(
year from shipping_limit_date) as year_2018, EXTRACT(month from shipping_limit_date) as mon
th_of_order, price + freight_value as cost from `Ecommerce.order_items`
where Extract(year from shipping_limit_date) = 2018)
group by month_of_order order by month_of_order)
```

```
SELECT co.month_of_order, round(cf.cost_2017,2) as cost_of_orders_2017,
round(co.cost_2018,2) as cost_of_orders_2018,
round((cost_2018-cost_2017)*100/cost_2017,2) as percentage_increase_month_on_month
FROM cost_of_orders_2017 cf join cost_of_orders_2018 co on
cf.month_of_order = co.month_of_order
where co.month_of_order between 1 and 8 order by co.month_of_order;
```

Result:

Row	month_of_order	cost_of_orders_2017	cost_of_orders_2018	percentage_increase_month_on_month
1	1	92198.41	987044.01	970.57
2	2	282731.52	958197.45	238.91
3	3	396542.39	1209226.26	204.94
4	4	352845.76	1116664.98	216.47
5	5	586221.79	1252698.09	113.69
6	6	543623.12	1021381.12	87.88
7	7	544850.17	964848.75	77.09
8	8	656244.33	1259175.95	91.88

In the above result, we can see that maximum increase in cost of orders occurred in the month of January.

Q2: Mean & Sum of price and freight value by customer state.

Query:

```
select customer_state, round(avg(price),2) as avg_price,round(sum(price),2) as total_price,
round(avg(freight_value),2) as avg_freight_value,
round(sum(freight_value),2) as total_freight_value,
round(avg(price + freight_value),2) as avg_cost,
round(sum(price + freight_value),2) as total_cost
from `Ecommerce.order_items` oi join `Ecommerce.orders` ord on oi.order_id = ord.order_id
join `Ecommerce.customers` cust on ord.customer_id = cust.customer_id
group by customer_state;
```

Result:

Row	customer_state	avg_price	total_price	avg_freight_value	total_freight_value	avg_cost	total_cost
1	SP	109.65	5202955.05	15.15	718723.07	124.8	5921678.12
2	RJ	125.12	1824092.67	20.96	305589.31	146.08	2129681.98
3	PR	119.0	683083.76	20.53	117851.68	139.54	800935.44
4	SC	124.65	520553.34	21.47	89660.26	146.12	610213.6
5	DF	125.77	302603.94	21.04	50625.5	146.81	353229.44
6	MG	120.75	1585308.03	20.63	270853.46	141.38	1856161.49
7	PA	165.69	178947.81	35.83	38699.3	201.53	217647.11
8	BA	134.6	511349.99	26.36	100156.68	160.97	611506.67
9	GO	126.27	294591.95	22.77	53114.98	149.04	347706.93
10	RS	120.34	750304.02	21.74	135522.74	142.07	885826.76

Part 5

Analysis on sales, freight and delivery time

Q1,2,3: Create columns to calculate average days between purchasing, delivering and estimated delivery

Query to create view:

```
CREATE OR REPLACE VIEW
Ecommerce.state_wise_analysis AS
SELECT
customer_state,
ROUND(AVG(freight_value),2) AS mean_freight_value,
ROUND(AVG(DATE_DIFF(order_delivered_carrier_date,order_purchase_timestamp,day)),2)
AS mean_time_to_delivery,
ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_carrier_date,day)),2) AS
mean_diff_estimated_delivery
FROM `Ecommerce.order_items` oi JOIN `Ecommerce.orders` ord ON oi.order_id = ord.order_id
JOIN `Ecommerce.customers` cust ON ord.customer_id = cust.customer_id
GROUP BY customer_state;
```

Q4: Sort the data:

(i): Top 5 states with lowest average freight value

Query: (Please run the view query first)

```
SELECT
customer_state,
mean_freight_value
FROM Ecommerce.state_wise_analysis ORDER BY mean_freight_value LIMIT 5;
```

Result:

Row	customer_state	mean_freight_value
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

We can see here that minimum freight value is of Sao Paulo (SP) followed by Parana (PR), Minas Gerais (MG), Rio De Janeiro (RJ).

(ii): Top 5 states with highest average freight value

Query:

```
SELECT
    customer_state,
    mean_freight_value
FROM Ecommerce.state_wise_analysis ORDER BY mean_freight_value DESC LIMIT 5;
```

Result:

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

(iii): Top 5 states with lowest average time(days) to delivery

Query:

```
SELECT
    customer_state,
    mean_time_to_delivery
FROM Ecommerce.state_wise_analysis ORDER BY mean_time_to_delivery LIMIT 5;
```

Result:

Row	customer_state	mean_time_to_delivery
1	AM	2.29
2	RO	2.34
3	GO	2.62
4	MS	2.72
5	MT	2.72

We can see in the above result that the fastest delivery happens in Amazonas (AM) state.

(iv): Top 5 states with highest average time to delivery

Query:

```
SELECT
    customer_state,
    mean_time_to_delivery
FROM Ecommerce.state_wise_analysis ORDER BY mean_time_to_delivery DESC LIMIT 5;
```

Result:

Row	customer_state	mean_time_to_delivery
1	RR	4.63
2	MA	3.4
3	SE	3.25
4	RN	3.2
5	AL	3.15

We can see in the above result that the slowest delivery happens in Roraima (RR) state.

(v): Top 5 states where delivery is not so fast compared to estimated date

Query:

```
SELECT
    customer_state,
    mean_diff_estimated_delivery
FROM Ecommerce.state_wise_analysis ORDER BY mean_diff_estimated_delivery LIMIT 5;
```

Result:

Row	customer_state	mean_diff_estimated_delivery
1	SP	15.68
2	DF	20.88
3	MG	21.0
4	PR	21.06
5	ES	21.8

(vi): Top 5 states where delivery is really fast compared to estimated date

Query:

```
SELECT
    customer_state,
    mean_diff_estimated_delivery
FROM Ecommerce.state_wise_analysis ORDER BY mean_diff_estimated_delivery DESC LIMIT 5;
```

Result:

Row	customer_state	mean_diff_estimated_delivery
1	AM	42.29
2	AP	42.0
3	RR	40.75
4	AC	37.3
5	RO	35.86

Part 6

Payment type analysis

Q1: Month over Month count of orders for different payment types

Query: `SELECT payment_type, month_of_order, count(distinct order_id) as no_of_orders FROM (SELECT payment_type, EXTRACT(MONTH FROM shipping_limit_date) as month_of_order, p.order_id FROM `Ecommerce.payments` p JOIN `Ecommerce.order_items` oi ON p.order_id = oi.order_id) GROUP BY payment_type, month_of_order ORDER BY payment_type, month_of_order;`

Result:

Row	payment_type	month_of_order	no_of_orders
1	UPI	1	1513
2	UPI	2	1600
3	UPI	3	1991
4	UPI	4	1675
5	UPI	5	2225
6	UPI	6	1722
7	UPI	7	1945
8	UPI	8	2389
9	UPI	9	879
10	UPI	10	1011
11	UPI	11	1191
12	UPI	12	1484
13	credit_card	1	5475

Query for total no. of orders by each payment type:

`SELECT payment_type, COUNT(DISTINCT p.order_id) AS no_of_orders FROM `Ecommerce.payments` p JOIN `Ecommerce.order_items` oi ON p.order_id = oi.order_id GROUP BY payment_type ORDER BY payment_type;`

Result:

Row	payment_type	no_of_orders
1	UPI	19614
2	credit_card	75991
3	debit_card	1521
4	voucher	3766

We can see from the above result that credit card is the preferred option for payment of Brazilians.

Q2: Distribution of payment instalments and count of orders

Query: `SELECT payment_installments, count(distinct order_id) as count_of_orders
FROM `Ecommerce.payments` GROUP BY payment_installments;`

Result:

Row	payment_installments	count_of_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644

From the above table, it can be inferred that most payments are done in one go.

Query for value of orders:

`SELECT payment_type, round(sum(payment_value),2) as value_of_orders
FROM `Ecommerce.payments` GROUP BY payment_type ORDER BY sum(payment_value) desc;`

Result:

Row	payment_type	value_of_orders
1	credit_card	12542084.19
2	UPI	2869361.27
3	voucher	379436.87
4	debit_card	217989.79
5	not_defined	0.0

Here we can see that Target gets the maximum revenue from payment type credit card only.

Query for value of orders state-wise:

```
SELECT customer_state, round(sum(payment_value),2) as value_of_orders FROM
`Ecommerce.payments` p JOIN `Ecommerce.orders` ord ON p.order_id = ord.order_id
JOIN `Ecommerce.customers` cust ON ord.customer_id = cust.customer_id
group by customer_state order by sum(payment_value) desc;
```

Result:

Row	customer_state	value_of_orders
1	SP	5998226.96
2	RJ	2144379.69
3	MG	1872257.26
4	RS	890898.54
5	PR	811156.38
6	SC	623086.43
7	BA	616645.82
8	DF	355141.08
9	GO	350092.31
10	ES	325967.55

Sao Paulo (SP) is the highest revenue generator for Target followed by Rio De Janeiro (RJ) and Minas Gerais (MG).

Insights	Recommendations
27 states (Actually 26 states and 1 federal district) and 4119 cities are covered by Target	All states and most of the cities have been covered by Target. And, more cities and small regions can be added to keep this growth in positive.
The analysis shows that maximum orders are placed in 8th Month (August) and then 2nd highest in 5th Month (May)	Target should aim month of august and month of may to increase their sales
Brazilian customers tend to buy in the afternoon that is from 12PM to 6PM	These hours can be aimed by Target to lure customers to buy
Highest no of orders is being placed in State & City Sao Paulo which is the financial centre of Brazil and State & City Rio De Janeiro is 2 nd which is the capital of Brazil.	Sao Paulo and Rio De Janeiro are the cities/states to concentrate more on for Target Considering the population of Sao Paulo and Rio De Janeiro, there is more potential to aim customers in these cities
Highest no of active & registered consumers is also in the state of Sao Paulo (SP) followed by Rio De Janeiro (RJ) and Minas Gerais (MG)	
Sao Paulo (SP) is the highest revenue generator for Target followed by Rio De Janeiro (RJ) and Minas Gerais (MG)	
The % increase in cost of orders from 2017 to 2018 is 153.79%	Increasing trend in cost of orders also indicates the increasing trend of E-commerce in Brazil. Giving Target more potential to grow.
Target gets the maximum no of orders & revenue from payment type credit card only	As more revenue and orders are coming from credit card, it could be beneficial to offer incentives on this payment type