

Question

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

1.1: Data type of all columns in the "customers" table.

Ans:

```
select column_name, data_type from Terget.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers'
```

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Inference:

There are two types of data types available in customers table. STRING and INT64

Question:

1.2: Get the time range between which the orders were placed.

Ans:

```
select min(order_purchase_timestamp) as first_day,
max(order_purchase_timestamp) as last_day
from `Terget.orders`
```

Row	first_day	last_day
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Inference:

The orders were placed between 2016-09-04 21:15:19 UTC and 2018-10-17 17:30:18UTC.

Question:

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

Ans:

```
select
count(distinct customer_city) as city_count,
count(distinct customer_state) as state_count
from `Terget.customers`
```

Row	city_count	state_count
1	4119	27

Inference:

There are 4119 unique cities and 27 states in the dataset.

Question:

2: In-depth Exploration:

2.1 : Is there a growing trend in the no. of orders placed over the past years?

Ans:

```
select
extract(month from order_purchase_timestamp) as month_order,
extract(year from order_purchase_timestamp) as year_order,
count(*) as order_count
from `Terget.orders`
group by year_order, month_order
order by year_order, month_order asc
```

Row	month_order	year_order	order_count
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404

Inference:

Over the past years the no of orders has increased significantly.

Question:

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

Ans:

```
with mainTable as (  
  select  
    extract(month from order_purchase_timestamp) as month_order,  
    extract(year from order_purchase_timestamp) as year_order,  
    count(order_id) as order_count  
  from `Terget.orders`  
  group by year_order, month_order  
  order by year_order, month_order asc  
)  
order_count_avg as  
(  
  select *, round(avg(order_count) over(partition by year_order),2) as avg_orderCount  
  from mainTable  
)  
Select * from order_count_avg  
where order_count > avg_orderCount  
  order by year_order, month_order asc
```

Row	month_order	year_order	order_count	avg_orderCount
1	10	2016	324	109.67
2	7	2017	4026	3758.42
3	8	2017	4331	3758.42
4	9	2017	4285	3758.42
5	10	2017	4631	3758.42
6	11	2017	7544	3758.42
7	12	2017	5673	3758.42
8	1	2018	7269	5401.1
9	2	2018	6728	5401.1

Inference:

From July to December in 2017, the number of orders is more than the average orders of 2017

Question:

2.3: 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

Ans:

```
with base as
(
    select
        extract(hour from order_purchase_timestamp) as hour_,
        count(order_id) as order_count
    from `Terget.orders`
    group by 1
),
DayTime as(
select *, case when hour_ between 0 and 6 then 'Dawn'
               when hour_ between 7 and 12 then 'Morning'
               when hour_ between 13 and 18 then 'After noon'
               when hour_ between 19 and 23 then 'Night'
               end as time_of_day
from base
)
select time_of_day, sum(order_count) as orders
from DayTime
group by 1
```

Row	time_of_day	orders
1	Morning	27733
2	Dawn	5242
3	After noon	38135
4	Night	28331

Inference:

During Afternoon, do the Brazilian customers mostly place their Orders.

Question:

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

3.1: Get the month on month no. of orders placed in each state.

Ans:

```
with base as
(
    select c.customer_id, o.order_id, o.order_purchase_timestamp, c.customer_state
    from `Terget.orders` as o
    inner join `Terget.customers` as c
    on o.customer_id=c.customer_id
)
select count(order_id) as orders, extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp) as year,
customer_state
from base
group by year , month, customer_state
order by year , month, customer_state
```

Row	orders	month	year	customer_state
1	1	9	2016	RR
2	1	9	2016	RS
3	2	9	2016	SP
4	2	10	2016	AL
5	4	10	2016	BA
6	8	10	2016	CE
7	6	10	2016	DF
8	4	10	2016	ES

Inference:

In September 2016, there is only one order placed from State "RR".

In October 2016, there are 8 orders placed from state "CE"

And so no.

Question:

3.2: How are the customers distributed across all the states?

Ans:

```
with base as
(
    select c.customer_id, o.order_id, o.order_purchase_timestamp, c.customer_state
    from `Terget.orders` as o
    inner join `Terget.customers` as c
    on o.customer_id=c.customer_id
)
select count(distinct customer_id) as customer, customer_state
```

```

from base
group by customer_state
order by customer desc

```

Row	customer	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

Inference:

There are 41746 customers from the state "SP",
 12852 customers from the state "RJ",
 11635 customers from the state "MG"
 And so on.

Question:

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

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

Ans:

```

with
mainT as
(
  SELECT extract(year from order_purchase_timestamp) as years,
  extract(month from order_purchase_timestamp) as months,
  sum(p.payment_value) as payments_
  from `Terget.orders` as o
  inner join `Terget.payments` as p
  on o.order_id=p.order_id
  where extract(month from order_purchase_timestamp) between 1 and 8
  group by 1, 2

```

```

        order by 1, 2
    ),
    grouped as
    (
        select years, sum(payments_) as payments from mainT
        group by 1
        order by 1
    ),
    nextYear as
    (
        select *, lead(grouped.payments,1) over (order by years asc) as next_year_payment
        from grouped
    )
SELECT *, round(((next_year_payment-payments)/payments)*100,2) as percentage_increase
from nextYear

```

Row	years	payments	next_year_payment	percentage_increase
1	2017	3669022.119999...	8694733.839999...	136.98
2	2018	8694733.839999...	null	null

Inference:

The % has increased in the cost of order from year 2017 to 2018 by 136.98%.

Question:

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

Ans:

```

with base as
(
    select c.customer_state, o.order_id, p.payment_value
    from `Terget.orders` as o
    inner join
    `Terget.customers` as c
    on o.customer_id=c.customer_id
    inner join
    `Terget.payments` as p
    on o.order_id=p.order_id
)
select customer_state, count(order_id) as orderCount, round(sum(payment_value),2) as
TotalPayment,
round(avg(payment_value),2) as avgPayment
from base
group by 1

```

Row	customer_state ▼	orderCount ▼	TotalPayment ▼	avgPayment ▼
1	RJ	13527	2144379.69	158.53
2	RS	5668	890898.54	157.18
3	SP	43622	5998226.96	137.5
4	DF	2204	355141.08	161.13
5	PR	5262	811156.38	154.15
6	MT	958	187029.29	195.23
7	MA	767	152523.02	198.86
8	AL	427	96962.06	227.08

Inference:

The total payment and average payment of orders in the state of "RJ" are "2144379.69" and "158.53" respectively. So no.

Question:

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

Ans:

```
with base as
(
    select c.customer_state, o.order_id, p.freight_value
    from `Terget.orders` as o
    inner join
    `Terget.customers` as c
    on o.customer_id=c.customer_id
    inner join
    `Terget.order_items` as p
    on o.order_id=p.order_id
)
select customer_state, count(order_id) as orderCount, round(sum(freight_value),2) as
Total_freight,
round(avg(freight_value)) as avg_freight
from base
group by 1
```


Row	customer_state ▼	orderCount ▼	Total_freight ▼	avg_freight ▼
1	MT	1055	29715.43	28.0
2	MA	824	31523.77	38.0
3	AL	444	15914.59	36.0
4	SP	47449	718723.07	15.0
5	MG	13129	270853.46	21.0
6	PE	1806	59449.66	33.0
7	RJ	14579	305589.31	21.0
8	DF	2406	50625.5	21.0

Inference:

The total freight value and average freight value of orders in the state of “MT” are “29715.43” and “28” Respectively. So no.

Question:

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

5.1: 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.

Ans:

```
select
order_id, customer_id, order_status,
order_purchase_timestamp, order_delivered_customer_date, order_estimated_delivery_date,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_purchase_timestamp, DAY) AS
time_to_deliver,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date, DAY) AS
diff_estimated_delivery
from `Terget.orders`
Where order_status="delivered"
```

Row	order_id	customer_id	order_status	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_delivery
1	cec8f5f7a13...	6be61d704faaf...	delivered	2017-03-17 15:56:47 UTC	2017-04-07 13:14:56 UTC	2017-05-18 00:00:00 UTC	61	40
2	58527ee472...	b7d68eb92ede5...	delivered	2017-03-20 11:01:17 UTC	2017-03-30 14:04:04 UTC	2017-05-18 00:00:00 UTC	58	48
3	10ed5499d1...	2bf569d940353...	delivered	2017-03-21 13:38:25 UTC	2017-04-18 13:52:43 UTC	2017-05-18 00:00:00 UTC	57	29
4	818996ea24...	19b1122a589ca...	delivered	2018-08-20 15:56:23 UTC	2018-08-29 22:52:40 UTC	2018-10-04 00:00:00 UTC	44	35
5	d195cac9cc...	a3a156d272fd0...	delivered	2018-08-12 18:14:29 UTC	2018-08-23 02:08:44 UTC	2018-10-04 00:00:00 UTC	52	41
6	64eeb35d3a...	d00827c5fac20...	delivered	2018-08-16 07:55:32 UTC	2018-08-23 00:09:45 UTC	2018-10-04 00:00:00 UTC	48	41
7	2691ae869f...	e551bab5d422f...	delivered	2018-08-22 22:39:54 UTC	2018-08-29 19:11:48 UTC	2018-10-04 00:00:00 UTC	42	35
8	1cd147d1c0...	b28dc057a0489...	delivered	2018-08-20 17:04:34 UTC	2018-08-29 16:41:59 UTC	2018-10-04 00:00:00 UTC	44	35

Inference:

For the first order the time to deliver the order is 61 days and difference of estimated deliver time 40 days. And so no.

Question:

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

Ans:

```
with base as
(
    select c.customer_state, o.order_id, p.freight_value
    from `Terget.orders` as o
    inner join
    `Terget.customers` as c
    on o.customer_id=c.customer_id
    inner join
    `Terget.order_items` as p
    on o.order_id=p.order_id
),
avg_freight_per_state as
(
    select customer_state,
    round(avg(freight_value)) as avg_freight
    from base
    group by 1
)
(
    Select customer_state, avg_freight from avg_freight_per_state
    order by avg_freight desc
    limit 5 )
union all
(Select customer_state, avg_freight from avg_freight_per_state
    order by avg_freight asc
    limit 5 )
```

Row	customer_state ▼	avg_freight ▼
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0
6	SP	15.0
7	PR	21.0
8	RJ	21.0
9	DF	21.0
10	MG	21.0

Inference:

Top 5 rows are the top 5 states with highest average freight value and last 5 rows are the 5 states with lowest average freight value.

5.3: Find out the top 5 states with the highest & lowest average delivery time

Ans:

```

with delivery_time_tab as(
select
o.order_id, o.order_status, o.order_delivered_customer_date, cT.customer_state as state,
o.order_estimated_delivery_date,
TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date, DAY) AS
delivery_time
from `Terget.orders` as o
inner join `Terget.customers` as cT
on o.customer_id=cT.customer_id
where
o.order_delivered_customer_date IS NOT NULL
)
(
select state,
avg(delivery_time) as avg_deliveryTime
from delivery_time_tab
group by 1
order by avg_deliveryTime desc
limit 5)
union all
(
select state,

```

```

avg(delivery_time) as avg_deliveryTime
from delivery_time_tab
group by 1
order by avg_deliveryTime asc
limit 5)

```

Row	state	avg_deliveryTime
1	AL	-7.94710327455...
2	MA	-8.76847977684...
3	SE	-9.17313432835...
4	ES	-9.61854636591...
5	BA	-9.93488943488...
6	AC	-19.7625000000...
7	RO	-19.1316872427...
8	AP	-18.7313432835...
9	AM	-18.6068965517...
10	RR	-16.4146341463...

Inference:

Top 5 rows are the top 5 states with highest average delivery time and last 5 rows are the 5 states with lowest average delivery time.

Question:

5.4: Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Ans:

```

with delivery_time_tab as(
select
o.order_id, o.order_status, o.order_delivered_customer_date, cT.customer_state as state,
o.order_estimated_delivery_date,
TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date, DAY) AS
delivery_time
from `Terget.orders` as o
inner join `Terget.customers` as cT

```

```

on o.customer_id=cT.customer_id
Where
    o.order_delivered_customer_date IS NOT NULL
)
(
select state,
    sum(delivery_time) as total_deliveryTime
from delivery_time_tab
group by 1
order by total_deliveryTime asc
limit 5)

```

Row	state	total_deliveryTime
1	SP	-410430
2	MG	-139632
3	RJ	-134667
4	RS	-69375
5	PR	-60869

Inference:

These are the top 5 states where the order delivery is superfast as compared to estimated date of delivery.

Question:

6. Analysis based on the payments:

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

Ans:

```

select p.payment_type,
    extract(month from o.order_purchase_timestamp) as months_,
    extract(year from o.order_purchase_timestamp) as years_, count(p.order_id) as orders_
from `Terget.payments` as p
inner join `Terget.orders` as o
on p.order_id=o.order_id
group by payment_type, years_, months_
order by years_, months_

```

Row	payment_type	months_	years_	orders_
1	credit_card	9	2016	3
2	credit_card	10	2016	254
3	voucher	10	2016	23
4	debit_card	10	2016	2
5	UPI	10	2016	63
6	credit_card	12	2016	1
7	voucher	1	2017	61
8	UPI	1	2017	197

Inference:

In September 2016, there are 3 orders placed by credit card,

In October 2016, there are 254 orders placed by credit card, and so on.

Question:

6.2: Find the no. of orders placed on the basis of the payment installments that have been paid.

Ans:

```
select payment_installments,
count(order_id) as no_of_orders from `Target.payments`
where payment_installments <> 0
group by payment_installments
```

Row	payment_installment	no_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920

Inference: with one payment installment there are 52546 orders in the table.
with 2 payment installment there are 12413 orders.. and so no.

