TARGET: SQL

Overview:

```
select * from customers;
select * from geolocation;
select * from order_items;
select * from order_reviews;
select * from orders;
select * from payments;
select * from products;
select * from sellers;
```

#1.a Data type of all columns in the “customers” table.

```
select
column_name,  data_type
from
information_schema.columns
where table_name = 'customers';
```

#1.b Get the time range between which the orders were placed.

```
select
min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order
from orders;
```

#1.c. Count the Cities & States of customers who ordered during the given period.

```
select
count(distinct customer_city) as cities,
count(distinct customer_state) as states
from customers;
```

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

```
select
extract(year from order_purchase_timestamp) as yr,
count(*) as order_count
from orders
group by yr
order by yr;
```

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

```
select
extract(year from order_purchase_timestamp) as yr,
extract(month from order_purchase_timestamp) as mnth,
count(*)
from orders
group by 1, 2
order by 1, 2;
```

#2.c. 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

```
select
case
 when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
 when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Morning'
 when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
 when extract(hour from order_purchase_timestamp) between 19 and 23 then 'Night'
end as time_of_day,
count(*) as num_order
from orders
group by 1
order by 2 desc;
```

#3.a. Get the month on month no. of orders placed in each state.

```
select
count(*) as num_order,
extract(month from order_purchase_timestamp) as mnth,
customer_state
from orders join customers using(customer_id)
group by 2, 3
order by 1;
```

#4.a. 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.

```
with base_1 as (
select * from orders o
join payments p using(order_id)
where extract(year from order_purchase_timestamp) between 2017 and 2018
and extract(month from order_purchase_timestamp) between 1 and 8
),
base_2 as (
select
extract(year from order_purchase_timestamp) as yr,
round(sum(payment_value), 2) as cost
from base_1
group by yr
order by 1
),
base_3 as (
select *,
lead(cost) over(order by yr) as nxt_year
from base_2
)
select *,
(nxt_year - cost) / cost * 100 as percent_increase
```

```
from base_3;

#4.b. Calculate the Total & Average value of order price for each state.
select
customer_state,
round(sum(price), 2) as total_price,
round(avg(price), 2) as avg_price
from order_items
join orders using(order_id)
join customers using(customer_id)
group by customer_state
order by 2;

#4.c. Calculate the Total & Average value of order freight for each state.
select
customer_state,
round(sum(freight_value), 2) as total_freight,
round(avg(freight_value), 2) as avg_freight
from order_items
join orders using(order_id)
join customers using(customer_id)
group by customer_state
order by 3;

#5.a. 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.
select order_id,
timestampdiff(day, order_delivered_customer_date, order_purchase_timestamp) as delivery_time,
timestampdiff(day, order_delivered_customer_date, order_estimated_delivery_date) as diff_estimated_delivery
from orders
where order_status = 'delivered';
```

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

```
(
select
customer_state,
round(avg(freight_value), 2) as avg_freight_value
from orders
join order_items using(order_id)
join customers using(customer_id)
group by customer_state
order by 2 desc
limit 5
)
union
(
select
customer_state,
round(avg(freight_value), 2) as avg_freight_value
from orders
join order_items using(order_id)
join customers using(customer_id)
group by customer_state
order by 2
limit 5
);
```

#5.d. 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.

```
select
customer_state,
round(sum(timestampdiff(day, order_delivered_customer_date, order_purchase_timestamp) /
count(order_id)), 2) as avg_del_time,
```

```
round(sum(timestampdiff(day, order_estimated_delivery_date, order_purchase_timestamp) /
count(order_id)), 2) as avg_est_del_time

from orders join customers using(customer_id)

where order_status = 'delivered'

group by customer_state

order by (avg_del_time - avg_est_del_time);
```

#6.a. Find the month on month no. of orders placed using different payment types.

```
select

payment_type,

extract(year from order_purchase_timestamp) as yr,

extract(month from order_purchase_timestamp) as mnth,

count(order_id) as num_order

from orders

join payments using(order_id)

group by payment_type, yr, mnth

order by 2, 3;
```

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

```
select

count(distinct order_id) as num_order,

payment_installments

from payments

where payment_installments >= 1

group by payment_installments;
```