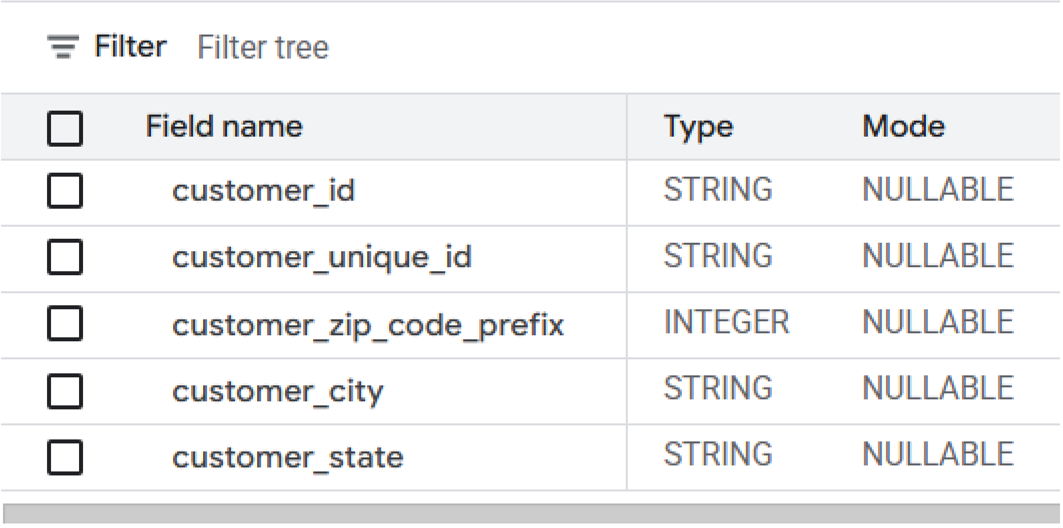
--Target project

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

--Data type of all columns in the "customers" table.



Filter Filter tree

□	Field name	Type	Mode
□	customer_id	STRING	NULLABLE
□	customer_unique_id	STRING	NULLABLE
□	customer_zip_code_prefix	INTEGER	NULLABLE
□	customer_city	STRING	NULLABLE
□	customer_state	STRING	NULLABLE

--Get the time range between which the orders were placed.

```
select order_id,
min(order_purchase_timestamp)as first_timestamp,
max(order_purchase_timestamp)as last_tmestamp
from `Target_project.orders`
group by order_id;
```

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

```
select count(distinct c.customer_city )as count_cities,
count(distinct c.customer_state)as count_states ,
min(o.order_purchase_timestamp)as starting_date,
max(o.order_purchase_timestamp)as ending_date
from `Target_project.customers` c
inner join `Target_project.orders` o
on c.customer_id=o.customer_id;
```

```sql
--In-depth Exploration:
--Is there a growing trend in the no. of orders placed over the past years?
with extract_year as(
select extract(year from o.order_purchase_timestamp )as year,
count(*)as total_orders
from `Target_project.orders` o
group by extract(year from o.order_purchase_timestamp ))

select year , total_orders,
lag(total_orders) over(order by year) as past_year,
total_orders - lag(total_orders) over(order by year)as diff_year,
case
  when total_orders >lag(total_orders) over(order by  year) then 'Growthing'
  when total_orders <lag(total_orders) over(order by  year) then 'Delining'
  else 'standed' end as Growing_trend
from extract_year
group by 1,2
order by 1,2;

--Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
select
extract(year from o.order_purchase_timestamp) as year,
format_date('%B',date(order_purchase_timestamp))as monthly,
extract(month from o.order_purchase_timestamp)as month_num,
count(*)as total_orders
from `Target_project.orders` o
group by monthly,year,month_num
order by year,monthly,month_num;
```

```
--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  c.customer_city,o.order_id,count(*)as num_orders,

case

   when extract(hour from o.order_purchase_timestamp) between 0 and 6 then 'Dawn'

   when extract(hour from o.order_purchase_timestamp) between 7 and 12 then 'Mornings'

   when extract(hour from o.order_purchase_timestamp) between 13 and 18 then 'afternoon'

   when extract(hour from o.order_purchase_timestamp) between 19 and 23 then 'night'

   end as day_time_order

from `Target_project.customers` c

inner join `Target_project.orders` o

on c.customer_id=o.customer_id

where upper( c.customer_city) = 'BRASILEIA'

group by 1,2,4;
```

```
--Evolution of E-commerce orders in the Brazil region:

--Get the month on month no. of orders placed in each state.
with cte as (
select
count(*) as total_orders,
extract(month from o.order_purchase_timestamp) as month,
c.customer_state
from `Target_project.customers` c
inner join `Target_project.orders` o
on o.customer_id= c.customer_id
group by month,c.customer_state
)
select customer_state,
total_orders,
month,
total_orders - lag(total_orders) over(partition by customer_state order by month) as month_on_month
from cte
order by customer_state;
```

```
--How are the customers distributed across all the states?

select
customer_state,count(distinct customer_id) as num_customer
from `Target_project.customers`
group by 1;
```

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


--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 cte as(
select
    extract(year from o.order_purchase_timestamp) as year,
    extract(month from o.order_purchase_timestamp) as month,
    round(sum(p.payment_value),4)as total_cost
from `Target_project.orders`o
inner join `Target_project.payments` p
on o.order_id=p.order_id
where extract(year from o.order_purchase_timestamp) between 2017 and 2018 and extract(month from o.order_purchase_timestamp) between 1 and 8
group by 1,2
order by 1,2
)
-- start to
select
round(sum(case when year= 2017 then total_cost else 0 end),4)as total_2017,
round(sum(case when year= 2018 then total_cost else 0 end),4)as total_2018,
round(
safe_divide(sum(case when year=2018 then total_cost else 0 end)- sum(case when year= 2017 then total_cost else 0 end),
sum(case when year= 2017 then total_cost else 0 end)) *100 ,4)as  increace_percent
from cte;----End This line is not mine I refered from chatgpt
```

```
--Calculate the Total & Average value of order price for each state.

select c.customer_state,

round(sum(i.price),4)as total_price,

round(avg(i.price),4)as avg_price

from `Target_project.customers` c

inner join Target_project.orders o

on c.customer_id = o.customer_id

inner join Target_project.order_items i

on o.order_id = i.order_id

group by 1

;


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

select c.customer_state,

round( sum(i.freight_value),4)as total_freight,

round(avg(i.freight_value),4)as avg_freight

from `Target_project.customers` c

inner join Target_project.orders o

on c.customer_id=o.customer_id

inner join `Target_project.order_items` i

on o.order_id=i.order_id

group by 1

order by c.customer_state;
```

```sql
--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.

--You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
--time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
--diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date
select
date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as diff_time_taken_deliver,
date_diff(order_delivered_customer_date,order_estimated_delivery_date, day)as diff_IN_days
from `Target_project.orders`;

--Find out the top 5 states with the highest & lowest average freight value.
select customer_state, avg_freight,highest_avg,lowest_avg
from(
select c.customer_state,avg(i.freight_value)as avg_freight,
dense_rank() over(order by avg(i.freight_value) desc)as highest_avg,
dense_rank() over(order by avg(i.freight_value)asc)as lowest_avg
from `Target_project.customers` c
inner join Target_project.orders o
on c.customer_id = o.customer_id
inner join `Target_project.order_items` i
on o.order_id = i.order_id
group by c.customer_state) sub
where highest_avg <= 5 or lowest_avg <=5
;
```

```
--Find out the top 5 states with the highest & lowest average delivery time.
with cte as(
select c.customer_state,
avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))as avg_delivery_time
from `Target_project.customers` c
inner join Target_project.orders o
on c.customer_id=o.customer_id
group by 1


)


 select customer_state, avg_delivery_time,
 case
 when highest<= 5 then 'Highest'
 else 'lowest' end as category
 from(
select
customer_state,avg_delivery_time,
row_number() over(order by avg_delivery_time desc)as highest,
row_number() over(order by avg_delivery_time asc)as lowest
from cte
 )
 where highest <= 5 or lowest <=5
 order by  avg_delivery_time
 ;
```

```
--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 c.customer_state,

avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day))as avg_early_delivery_date

from `Target_project.customers` c

inner join Target_project.orders o

on c.customer_id = o.customer_id

group by 1

order by avg_early_delivery_date desc

limit 5;



--Analysis based on the payments:

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

select

extract(year from o.order_purchase_timestamp)as year,

extract(month from o.order_purchase_timestamp)as month,

count(distinct o.order_id)as order_placed,

p.payment_type

from `Target_project.payments` p

inner join `Target_project.orders` o

on p.order_id = o.order_id

group by 1,2,p.payment_type

order by year,month;
```

```
--Find the no. of orders placed on the basis of the payment installments that have been paid.
select count(distinct o.order_id)as order_placed,p.payment_installments
from `Target_project.payments` p
inner join Target_project.orders o
on p.order_id = o.order_id
where p.payment_installments >0
group by p.payment_installments
;
```