BigQuery

```sql
-- -- Overview
select count(distinct customer_id) as no_of_customers
from `target_sql.customers`;


select count(distinct seller_id) as no_of_sellers
from `target_sql.sellers`;


select count(distinct order_id) as no_of_orders
from `target_sql.orders`;



-- -- Initial Exploration
select c.customer_id, c.customer_city, c.customer_state, o.order_id, o.order_purchase_timestamp
from `target_sql.orders` o
inner join `target_sql.customers` c
on o.customer_id = c.customer_id
where o.order_purchase_timestamp = (
 select min(order_purchase_timestamp)
 from `target_sql.orders`
);


select c.customer_id, c.customer_city, c.customer_state, o.order_id, o.order_purchase_timestamp
from `target_sql.orders` o
inner join `target_sql.customers` c
on o.customer_id = c.customer_id
where o.order_purchase_timestamp = (
 select max(order_purchase_timestamp)
 from `target_sql.orders`
);


select customer_city, count(customer_city) as Count
from `target_sql.customers`
group by customer_city;
```

```sql
select customer_state, count(customer_state) as Count
from `target_sql.customers`
group by customer_state;



-- -- In-depth Exploration
select t2.Quarter, count(t2.order_id) as No_of_orders from (
 select
   t1.order_id,
   t1.Month_Year_Name,
   case
     when (t1.month between 9 and 12) and t1.year = 2016 then '2016_Q4'
     when (t1.month between 1 and 3) and t1.year = 2017 then '2017_Q1'
     when (t1.month between 4 and 6) and t1.year = 2017 then '2017_Q2'
     when (t1.month between 7 and 9) and t1.year = 2017 then '2017_Q3'
     when (t1.month between 10 and 12) and t1.year = 2017 then '2017_Q4'
     when (t1.month between 1 and 3) and t1.year = 2018 then '2018_Q1'
     when (t1.month between 4 and 6) and t1.year = 2018 then '2018_Q2'
     when (t1.month between 7 and 10) and t1.year = 2018 then '2018_Q3'
   end as Quarter
 from (
   select
     order_id,
     extract(year from order_purchase_timestamp) as year, extract(month from
order_purchase_timestamp) as month,
     format_date('%B %Y', order_purchase_timestamp) as Month_Year_Name
   from `target_sql.orders`
 ) as t1
) as t2
group by t2.Quarter
order by t2.Quarter;



select t1.Quarter, t1.Month_Year_Name, t1.No_of_orders from (
 select
   count(order_id) as No_of_orders,
   extract(year from order_purchase_timestamp) as year, extract(month from
order_purchase_timestamp) as month,
   format_date('%B %Y', order_purchase_timestamp) as Month_Year_Name,
   case
     when (extract(month from order_purchase_timestamp) between 9 and 12) and extract(year from
order_purchase_timestamp) = 2016 then '2016_Q4'
```

```sql
     when (extract(month from order_purchase_timestamp) between 1 and 3) and extract(year from
order_purchase_timestamp) = 2017 then '2017_Q1'
     when (extract(month from order_purchase_timestamp) between 4 and 6) and extract(year from
order_purchase_timestamp) = 2017 then '2017_Q2'
     when (extract(month from order_purchase_timestamp) between 7 and 9) and extract(year from
order_purchase_timestamp) = 2017 then '2017_Q3'
     when (extract(month from order_purchase_timestamp) between 10 and 12) and extract(year from
order_purchase_timestamp) = 2017 then '2017_Q4'
     when (extract(month from order_purchase_timestamp) between 1 and 3) and extract(year from
order_purchase_timestamp) = 2018 then '2018_Q1'
     when (extract(month from order_purchase_timestamp) between 4 and 6) and extract(year from
order_purchase_timestamp) = 2018 then '2018_Q2'
     when (extract(month from order_purchase_timestamp) between 7 and 10) and extract(year from
order_purchase_timestamp) = 2018 then '2018_Q3'
   end as Quarter
 from `target_sql.orders`
 group by year, month, Month_Year_Name, Quarter
) as t1
order by t1.Quarter, t1.year, t1.month;


select
 case
   when extract(time from order_purchase_timestamp) between '00:00:00' and '05:59:59'
     then 'Night (12 AM to 6 AM)'
   when extract(time from order_purchase_timestamp) between '06:00:00' and '11:59:59'
     then 'Morning (6 AM to 12 PM)'
   when extract(time from order_purchase_timestamp) between '12:00:00' and '17:59:59'
     then 'Afternoon (12 PM to 6 PM)'
   else 'Evening (6 PM to 12 AM)'
 end as Time_Frame,
 count(order_id) as No_of_orders
from `target_sql.orders`
group by Time_Frame
order by No_of_orders desc;



-- -- Evolution of E-commerce
select t1.customer_state, t1.Month_Year_Name, count(t1.order_id) as No_of_orders from (
 select
   o.order_id, c.customer_state,
   extract(month from o.order_purchase_timestamp) as Month,
```

```sql
   extract(year from o.order_purchase_timestamp) as Year,
   format_date('%B %Y', order_purchase_timestamp) as Month_Year_Name
 from `target_sql.customers` c
 inner join `target_sql.orders` o
 on c.customer_id = o.customer_id) as t1
group by  t1.customer_state, t1.Year, t1.Month, t1.Month_Year_Name
order by t1.customer_state, t1.Year, t1.Month;


select
 distinct customer_state,
 count(customer_id) as No_of_customers
from `target_sql.customers`
group by customer_state
order by No_of_customers desc;


-- -- Impact on Economy
with total_payment_val_2017 as (
 select round(sum(t1.payment_value), 2) as total_payment_2017 from (
   select
     p.order_id, p.payment_value
   from `target_sql.orders` o join `target_sql.payments` p
   on o.order_id = p.order_id
   where extract(year from o.order_purchase_timestamp) = 2017 and extract(month from
o.order_purchase_timestamp) between 1 and 8
 ) as t1
),
total_payment_val_2018 as (
 select round(sum(t2.payment_value), 2) as total_payment_2018 from (
   select
     p.order_id, p.payment_value
   from `target_sql.orders` o join `target_sql.payments` p
   on o.order_id = p.order_id
   where extract(year from o.order_purchase_timestamp) = 2018 and extract(month from
o.order_purchase_timestamp) between 1 and 8
 ) as t2
)
select
 total_payment_2017, total_payment_2018,
 concat(round((((total_payment_2018 - total_payment_2017) / total_payment_2017) * 100), 2), '%')
as percent_increase_2017_to_2018
from total_payment_val_2017, total_payment_val_2018;
```

```sql
with total_payment_val_2017 as (
 select t1.month, t1.Month_Name, t1.month_wise_payment_2017 from (
   select
     round(sum(p.payment_value), 2) as month_wise_payment_2017,
     format_date('%B', o.order_purchase_timestamp) as Month_Name, extract(month from
o.order_purchase_timestamp) as month
   from `target_sql.orders` as o join `target_sql.payments` as p
   on o.order_id = p.order_id
   where extract(month from o.order_purchase_timestamp) between 1 and 8 and extract(year from
o.order_purchase_timestamp) = 2017
   group by month, Month_Name
   order by month
) as t1
),
total_payment_val_2018 as (
 select t2.Month_Name, t2.month_wise_payment_2018 from (
   select
     round(sum(p.payment_value), 2) as month_wise_payment_2018,
     format_date('%B', o.order_purchase_timestamp) as Month_Name, extract(month from
o.order_purchase_timestamp) as month
   from `target_sql.orders` as o join `target_sql.payments` as p
   on o.order_id = p.order_id
   where extract(month from o.order_purchase_timestamp) between 1 and 8 and extract(year from
o.order_purchase_timestamp) = 2018
   group by month, Month_Name
   order by month
 ) as t2
)
select
 t_2017.Month_Name,
 t_2017.month_wise_payment_2017,
 t_2018.month_wise_payment_2018,
 concat(round((((t_2018.month_wise_payment_2018 - t_2017.month_wise_payment_2017) /
t_2017.month_wise_payment_2017) * 100), 2), '%') as month_percent_increase_2017_to_2018
from total_payment_val_2017 as t_2017
join total_payment_val_2018 as t_2018
on t_2017.Month_Name = t_2018.Month_Name
order by t_2017.month;



select
 c.customer_state, count(oi.order_id) as Count,
 round(sum(oi.price), 2) as sum_of_price,
```

```sql
 round(avg(oi.price), 2) as mean_of_price,
 round(sum(oi.freight_value), 2) as sum_of_freight_value,
 round(avg(oi.freight_value), 2) as mean_of_freight_value
from `target_sql.customers` c
join `target_sql.orders` o
on c.customer_id = o.customer_id
join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by mean_of_price desc, mean_of_freight_value desc;



-- -- Analysis on sales, freight and delivery time
select
 order_id,
 extract(date from order_purchase_timestamp) as order_purchase,
 extract(date from order_delivered_customer_date) as order_delivered,
 extract(date from order_estimated_delivery_date) as order_estimated_delivery,
 date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
days_between_purchase_delivered,
 date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as
days_between_estimated_delivered
from `target_sql.orders`
where order_delivered_customer_date is not null
order by days_between_purchase_delivered desc;


select
 t1.order_id, t1.order_purchase, t1.order_delivered,
 date_diff(t1.order_delivered, t1.order_purchase, day) as time_to_delivery_in_days
from (
 select
   order_id,
   extract(date from order_purchase_timestamp) as order_purchase,
   extract(date from order_delivered_customer_date) as order_delivered
 from `target_sql.orders`
 where order_delivered_customer_date is not null
)as t1
order by time_to_delivery_in_days desc;
```

```sql
select
 t1.order_id, t1.order_estimated_delivery, t1.order_delivered,
 date_diff(t1.order_estimated_delivery, t1.order_delivered, day) as
diff_estimated_delivery_in_days
from (
 select
   order_id,
   extract(date from order_estimated_delivery_date) as order_estimated_delivery,
   extract(date from order_delivered_customer_date) as order_delivered
 from `target_sql.orders`
 where order_delivered_customer_date is not null
)as t1
order by diff_estimated_delivery_in_days desc;



select
 distinct c.customer_state,
 round(avg(oi.freight_value), 2) as mean_freight_value,
 concat(round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)),
2), ' days') as avg_time_to_delivery,
 concat(round(avg(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date,
day)), 2), ' days') as avg_diff_estimated_delivery
from `target_sql.customers` c
join `target_sql.orders` o
on c.customer_id = o.customer_id
join `target_sql.order_items` oi
on o.order_id = oi.order_id
where (o.order_delivered_customer_date, o.order_purchase_timestamp,
o.order_estimated_delivery_date) is not null
group by c.customer_state
order by mean_freight_value;

select
 distinct c.customer_state,
 round(avg(oi.freight_value), 2) as avg_freight_value,
from `target_sql.customers` c
join `target_sql.orders` o
on c.customer_id = o.customer_id
join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by avg_freight_value desc
limit 5;
```

```sql
select
 distinct c.customer_state,
 round(avg(oi.freight_value), 2) as avg_freight_value,
from `target_sql.customers` c
join `target_sql.orders` o
on c.customer_id = o.customer_id
join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by avg_freight_value
limit 5;


select t1.customer_state, t1.avg_time_to_delivery from (
 select
   distinct c.customer_state,
   concat(round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,
day)), 2), ' days') as avg_time_to_delivery,
   round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)), 2) as
order_by_time
 from `target_sql.customers` c
 join `target_sql.orders` o
 on c.customer_id = o.customer_id
 join `target_sql.order_items` oi
 on o.order_id = oi.order_id
 group by c.customer_state
 order by order_by_time
 limit 5
) as t1;


select t1.customer_state, t1.avg_time_to_delivery from (
 select
   distinct c.customer_state,
   concat(round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,
day)), 2), ' days') as avg_time_to_delivery,
   round(avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day)), 2) as
order_by_time
 from `target_sql.customers` c
 join `target_sql.orders` o
 on c.customer_id = o.customer_id
 join `target_sql.order_items` oi
```

```
 on o.order_id = oi.order_id
 group by c.customer_state
 order by order_by_time desc
 limit 5
) as t1;


select t1.customer_state, t1.avg_diff_estimated_delivery from (
 select
   distinct c.customer_state,
   concat(round(avg(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date,
day)), 2), ' days') as avg_diff_estimated_delivery,
   round(avg(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, day)),
2) as order_by_diff,
 from `target_sql.customers` c
 join `target_sql.orders` o
 on c.customer_id = o.customer_id
 join `target_sql.order_items` oi
 on o.order_id = oi.order_id
 group by c.customer_state
 order by order_by_diff
 limit 5
) as t1;


select t1.customer_state, t1.avg_diff_estimated_delivery from (
 select
   distinct c.customer_state,
   concat(round(avg(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date,
day)), 2), ' days') as avg_diff_estimated_delivery,
   round(avg(date_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, day)),
2) as order_by_diff,
 from `target_sql.customers` c
 join `target_sql.orders` o
 on c.customer_id = o.customer_id
 join `target_sql.order_items` oi
 on o.order_id = oi.order_id
 group by c.customer_state
 order by order_by_diff desc
 limit 5
) as t1;
```

```sql
-- -- Payment type analysis
select
 t1.payment_type, t1.Month_Year_Name, t1.No_of_orders,
 sum(t1.No_of_orders) over(partition by t1.payment_type order by t1.year, t1.month) as
month_over_month_count_of_orders
from (
 select
   count(o.order_id) as No_of_orders,
   p.payment_type,
   extract(year from o.order_purchase_timestamp) as year, extract(month from
o.order_purchase_timestamp) as month,
   format_date('%B %Y', o.order_purchase_timestamp) as Month_Year_Name
 from `target_sql.orders` o
 join `target_sql.payments` p
 on o.order_id = p.order_id
 group by year, month, Month_Year_Name, p.payment_type
 order by p.payment_type, year, month
) as t1;


select
 distinct payment_installments,
 count(order_id) as No_of_orders,
 round(sum(payment_value), 2) as total_payment
from `target_sql.payments`
group by payment_installments;



-- --Extra
select
 oi.order_id,
 oi.product_id,
 oi.price,
 oi.freight_value,
 p.product_category
from `target_sql.order_items` oi
join `target_sql.products` p
on oi.product_id = p.product_id
order by oi.freight_value desc;
```

```sql
select
 distinct payment_type,
 count(order_id) as no_of_orders,
 round(sum(payment_value), 2) as total_payment
from `target_sql.payments`
group by payment_type
order by no_of_orders desc;
```