```
/*
From, where, group by, having, select order by, limit
*/
SELECT *
FROM moon.customers
LIMIT 10;

SELECT *
FROM moon.order_items
LMIT 10;

SELECT *
FROM moon.orders
LIMIT 10;

SELECT *
FROM moon.products
LIMIT 10;

/*
Level of the Table
*/

SELECT COUNT(*) AS all_count FROM moon.order_items;

SELECT COUNT(*) FROM
(
SELECT order_id, order_item_id
FROM moon.order_items
GROUP BY 1,2
 );

/*
1.1  Data type of columns in a table
*/

DESCRIBE TABLE moon.customers;

/*
1.2 Time period for which the data is given
```

*/

SELECT
MIN(order_purchase_timestamp) AS first_order,
MAX(order_purchase_timestamp) AS last_order
FROM moon.orders;

/*
1.3 Number of cities and states in our dataset
*/

SELECT
COUNT(DISTINCT (geolocation_city)) AS city_count,
COUNT(DISTINCT (geolocation_state)) AS state_count
FROM moon.geolocations;

/*
2.1 Is there a growing trend in e-commerce in Brazil? How can we describe a complete scenario?
*/

SELECT
EXTRACT(year FROM timestamp(order_purchase_timestamp)) AS year,
EXTRACT(month FROM timestamp(order_purchase_timestamp)) AS month,
COUNT(1) AS num_orders
FROM moon.orders GROUP BY year, month ORDER BY year, month;

/*
2.1 second part
Can we see some seasonality with peaks at specific months?
*/

SELECT
EXTRACT(month FROM timestamp(order_purchase_timestamp)) AS month,
COUNT(1) AS num_orders
FROM moon.orders
GROUP BY 1
ORDER BY 1;

/*

2.2 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)
***/**

```
SELECT
CASE
        WHEN EXTRACT(hour FROM timestamp(order_purchase_timestamp))
BETWEEN 0
AND 6
            THEN 'dawn'
        WHEN EXTRACT(hour FROM timestamp(order_purchase_timestamp))
BETWEEN 7
AND 12
            THEN 'morning'
        WHEN EXTRACT(hour FROM timestamp(order_purchase_timestamp))
BETWEEN 13
AND 18
            THEN 'afternoon'
        WHEN EXTRACT(hour FROM timestamp(order_purchase_timestamp))
BETWEEN 19
AND 23
           THEN 'night'
END AS time_of_day,
COUNT(DISTINCT order_id) AS counter FROM moon.orders
GROUP BY 1
ORDER BY 2 DESC;
```

/*
3.1 Get month on month orders by region/ state.
***/**

```
SELECT
EXTRACT(month FROM timestamp(order_purchase_timestamp)) AS month,
g.geolocation_state,
COUNT(1) AS num_orders
FROM moon.orders o
INNER JOIN moon.customers c
ON o.customer_id = c.customer_id
INNER JOIN moon.geolocations g
ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
```

```
GROUP BY g.geolocation_state, month
ORDER BY geolocation_state DESC, month ASC;

/*
3.2 How are customers distributed in Brazil
*/

SELECT
g.geolocation_state,
COUNT(DISTINCT (c.customer_unique_id)) AS num_customers
FROM moon.customers c
INNER JOIN moon.geolocations g
ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
GROUP BY g.geolocation_state
ORDER BY num_customers DESC;



/*
4.1 Analyze the money movement by e-commerce by looking at order prices,
freight and others.
*/

WITH
cte_table AS (
SELECT
        EXTRACT(month FROM timestamp(o.order_purchase_timestamp)) AS month,
        EXTRACT(year FROM timestamp(o.order_purchase_timestamp)) AS year,
        (sum(price) / COUNT( distinct o.order_id)) AS price_per_order,
        (sum(freight_value) / COUNT(distinct o.order_id)) AS freight_per_order
FROM moon.orders o
INNER JOIN moon.order_items i
ON o.order_id = i.order_id
GROUP BY year, month
)

SELECT (price_per_order), (freight_per_order), month, year
FROM cte_table order by year asc, month asc ;
```

/*
4.2 Mean & Sum of price and freight value by customer state
*/

```
with cte_table as
(
SELECT
c.customer_state as state,
sum(price) as total_price,
sum(freight_per_order) as total_freight,
count(distinct(o.order_id)) as num_orders
FROM moon.orders o inner join moon.order_items i
on o.order_id= i.order_id inner join moon.customers c
on o.customer_id=c.customer_id group by state
)
select
state, total_price, num_orders,(total_price/num_orders) as avg_price
from cte_table order by total_price desc;
select
state, total_freight, num_orders,(total_freight/num_orders) as AverageFreight
from cte_table order by total_freight desc;
```

/*
5.1 and 5.2 Analysis on sales, freight and delivery time create new columns for time to delivery and difference in estimated vs actual delivery
*/

```
SELECT
order_id,
date_DIFF
(
date(order_estimated_delivery_date),
date(order_purchase_timestamp),
DAY
) AS time_to_del,
TIMESTAMP_DIFF(
timestamp(order_delivered_customer_date),
timestamp(order_estimated_delivery_date),
DAY) AS diff_estimated_dil
FROM moon.orders WHERE order_status = 'delivered';
```

/*

5.5 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

*/

```
SELECT , Avg(freight_per_order) AS AverageFreight
FROM moon.orders
GROUP BY year,month,
ORDER BY year,month DESC
LIMIT 5;
```

/*

5.6 Top 5 states where delivery is really fast/ not so fast compared to estimated date

*/

```
SELECT , Avg(time_to_delivery) AS EstimatedDelivery
FROM moon.orders
GROUP BY year,month,
ORDER BY year,month DESC
LIMIT 5;
```

/*

5.7 Top 5 states with highest/lowest average time to delivery

*/

```
SELECT g.geolocation_state as state,
SUM
(
TIMESTAMP_DIFF
(
timestamp(order_estimated_delivery_date),
timestamp(order_purchase_timest amp), DAY
)'
)/COUNT(ORDER_ID) AS avg_dil_time,
FROM moon.orders o
```

```
inner join moon.customers c
on o.customer_id=c.customer_id
inner join moon.geolocations g
on c.customer_zip_code_prefix=g.geolocation_zip_code_prefix
where order_status='delivered'
group by state
order by avg_dil_time limit 5;

/*
6.1 Month over Month count of orders for different payment types
*/

SELECT month, orders, lagger_orders,
(orders - coalesce(lagger_orders, 0)) / coalesce(lagger_orders, 1) * 100
AS difference
FROM (
 SELECT *, lag(orders, 1) OVER (ORDER BY month ASC) AS lagger_orders
FROM (
SELECT EXTRACT(month FROM timestamp(a.order_purchase_timestamp)) AS month,
COUNT(DISTINCT a.order_id) AS orders,
COUNT(DISTINCT b.customer_unique_id) AS customers
FROM moon.orders a
LEFT JOIN moon.customers b
ON a.customer_id = b.customer_id WHERE EXTRACT(payment_type FROM
timestamp(a.order_purchase_timestamp))
GROUP BY 1 )
base )
base_2
ORDER BY month ASC;
```