1.1. Data type of columns in a table

SELECT column\_name, data\_type

FROM `Customer.INFORMATION\_SCHEMA.COLUMNS`

WHERE table\_name = "Customer\_info";

1.2.Time period for which the data is given

SELECT MIN(extract(YEAR FROM order\_purchase\_timestamp)) as Start\_year\_data,MAX(extract(YEAR FROM order\_purchase\_timestamp)) as End\_year\_data FROM `first-business-case-study.Customer.orders`;

* 1. Cities and States of customers ordered during the given period

select distinct customer.customer\_city,customer.customer\_state

from `first-business-case-study.Customer.orders` orders

join `first-business-case-study.Customer.Customer\_info` customer on orders.customer\_id=customer.customer\_id

order by customer.customer\_city,customer.customer\_state;

2.1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

SELECT count(orders.order\_id) as cnt\_order,

extract(YEAR from orders.order\_purchase\_timestamp) as order\_year,

extract(MONTH from orders.order\_purchase\_timestamp) as order\_month

FROM `first-business-case-study.Customer.orders` orders

LEFT JOIN `first-business-case-study.Customer.Customer\_info` customer

ON orders.customer\_id = customer.customer\_id

group by order\_year,order\_month;

2.2.

3.1. Get month on month orders by states

SELECT extract(MONTH from orders.order\_purchase\_timestamp) as order\_month, count(\*) as cnt\_orders\_month, geolocation.geolocation\_state FROM `first-business-case-study.Customer.orders` orders

INNER JOIN `first-business-case-study.Customer.Customer\_info` cust

ON orders.customer\_id = cust.customer\_id

INNER JOIN `first-business-case-study.Customer.Geolocation` geolocation

ON cust.customer\_zip\_code\_prefix = geolocation.geolocation\_zip\_code\_prefix

group by order\_month,geolocation.geolocation\_state

order by order\_month;

3.2. Distribution of customers across the states in Brazil

SELECT customer\_state, count(\*) as cnt\_customer\_per\_state FROM `first-business-case-study.Customer.Customer\_info` group by customer\_state;

SELECT customer\_state, ROUND((count(\*)/(SELECT count(\*)

FROM `first-business-case-study.Customer.Customer\_info`))\*100,2) as cnt\_customer\_per\_state FROM `first-business-case-study.Customer.Customer\_info` group by customer\_state;

* 1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment\_value” column in payments table

WITH payment\_2017 as

(SELECT extract(YEAR from orders.order\_purchase\_timestamp) as orders\_year, extract(MONTH from orders.order\_purchase\_timestamp) as orders\_month, ROUND(SUM(payment.payment\_value),2) as payment\_amount FROM `first-business-case-study.Customer.payments` payment

LEFT JOIN `first-business-case-study.Customer.orders` orders

ON payment.order\_id = orders.order\_id

where extract(MONTH from orders.order\_purchase\_timestamp) between 1 and 8 AND extract(YEAR from orders.order\_purchase\_timestamp)= 2017

group by orders\_year, orders\_month

order by orders\_year, orders\_month),

payment\_2018 as

(SELECT extract(YEAR from orders.order\_purchase\_timestamp) as orders\_year\_next, extract(MONTH from orders.order\_purchase\_timestamp) as orders\_month\_next, ROUND(SUM(payment.payment\_value),2) as payment\_amount\_next FROM `first-business-case-study.Customer.payments` payment

LEFT JOIN `first-business-case-study.Customer.orders` orders

ON payment.order\_id = orders.order\_id

where extract(MONTH from orders.order\_purchase\_timestamp) between 1 and 8 AND extract(YEAR from orders.order\_purchase\_timestamp)= 2018

group by orders\_year\_next, orders\_month\_next

order by orders\_year\_next, orders\_month\_next)

select \*,

ROUND(((payment\_2018.payment\_amount\_next - payment\_2017.payment\_amount)/payment\_2017.payment\_amount)\*100,2) as percentage\_increase from payment\_2017

INNER JOIN payment\_2018

ON payment\_2017.orders\_month = payment\_2018.orders\_month\_next

order by payment\_2017.orders\_month;

* 1. Mean & Sum of price and freight value by customer state

SELECT customer.customer\_state, ROUND(avg(order\_item.price),2) as mean\_price, ROUND(sum(order\_item.price),2) as total\_price,

ROUND(avg(order\_item.freight\_value),2) as mean\_freight\_value, ROUND(sum(order\_item.freight\_value),2) as total\_freight\_value

FROM `first-business-case-study.Customer.order\_items` order\_item

INNER JOIN `first-business-case-study.Customer.orders` orders

ON order\_item.order\_id = orders.order\_id

INNER JOIN `first-business-case-study.Customer.Customer\_info` customer

ON orders.customer\_id = customer.customer\_id

group by customer.customer\_state

order by customer.customer\_state;

5.1. Calculate days between purchasing, delivering and estimated delivery

SELECT DATE\_DIFF(order\_delivered\_carrier\_date, order\_purchase\_timestamp, day) as purchase\_delivery\_day\_diff,

DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_carrier\_date, day) as estimated\_delivery\_day\_diff,

DATE\_DIFF(order\_estimated\_delivery\_date, order\_purchase\_timestamp, day) as estimated\_purchase\_day\_diff

FROM `first-business-case-study.Customer.orders`;

OR

SELECT

DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, day) as purchase\_delivery\_day\_diff,

DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day) as estimated\_delivery\_day\_diff,

DATE\_DIFF(order\_estimated\_delivery\_date, order\_purchase\_timestamp, day) as estimated\_purchase\_day\_diff

FROM `first-business-case-study.Customer.orders`;

5.2. Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:

* + time\_to\_delivery = order\_delivered\_customer\_date-order\_purchase\_timestamp
  + diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

SELECT DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp,day) as time\_to\_delivery,

DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day) as diff\_estimated\_delivery

FROM `first-business-case-study.Customer.orders`

where (order\_delivered\_customer\_date - order\_purchase\_timestamp) is not null or (order\_estimated\_delivery\_date - order\_delivered\_customer\_date) is not null;

5.3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

SELECT customer.customer\_state,

ROUND(AVG(order\_items.freight\_value),2) as mean\_freight\_value,

ROUND(AVG(DATE\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,day)),2) as time\_to\_delivery,

ROUND(AVG(DATE\_DIFF(orders.order\_estimated\_delivery\_date, orders.order\_delivered\_customer\_date, day)),2) as diff\_estimated\_delivery

FROM `first-business-case-study.Customer.orders` orders

LEFT JOIN `first-business-case-study.Customer.order\_items` order\_items

ON orders.order\_id = order\_items.order\_id

LEFT JOIN `first-business-case-study.Customer.Customer\_info` customer

ON orders.customer\_id = customer.customer\_id

group by customer.customer\_state

order by customer.customer\_state;

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

WITH State\_freight\_value as

(SELECT customer.customer\_state,

ROUND(AVG(order\_items.freight\_value),2) as mean\_freight\_value,

ROUND(AVG(DATE\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,day)),2) as time\_to\_delivery,

ROUND(AVG(DATE\_DIFF(orders.order\_estimated\_delivery\_date, orders.order\_delivered\_customer\_date, day)),2) as diff\_estimated\_delivery,

dense\_rank() over(order by ROUND(AVG(order\_items.freight\_value),2) desc) as max\_freight\_value,

dense\_rank() over(order by ROUND(AVG(order\_items.freight\_value),2) asc) as min\_freight\_value

FROM `first-business-case-study.Customer.orders` orders

LEFT JOIN `first-business-case-study.Customer.order\_items` order\_items

ON orders.order\_id = order\_items.order\_id

LEFT JOIN `first-business-case-study.Customer.Customer\_info` customer

ON orders.customer\_id = customer.customer\_id

group by customer.customer\_state),

max\_freight as

(SELECT customer\_state,mean\_freight\_value,max\_freight\_value FROM State\_freight\_value

WHERE max\_freight\_value<=5),

min\_freight as

(SELECT customer\_state,mean\_freight\_value,min\_freight\_value FROM State\_freight\_value

WHERE min\_freight\_value<=5)

SELECT max\_freight.customer\_state as top\_five\_state,max\_freight.mean\_freight\_value as top\_five\_freight\_value,

min\_freight.customer\_state as lower\_five\_state,min\_freight.mean\_freight\_value as lower\_five\_freight\_value

FROM max\_freight

INNER JOIN min\_freight

ON max\_freight.max\_freight\_value = min\_freight.min\_freight\_value

order by max\_freight.max\_freight\_value;

5.6. Top 5 states with highest/lowest average time to delivery

WITH State\_time\_to\_delivery as

(SELECT customer.customer\_state,

ROUND(AVG(DATE\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,day)),2) as time\_to\_delivery,

dense\_rank() over(order by ROUND(AVG(DATE\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,day)),2) desc) as max\_time\_delivery,

dense\_rank() over(order by ROUND(AVG(DATE\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,day)),2) asc) as min\_time\_delivery

FROM `first-business-case-study.Customer.orders` orders

LEFT JOIN `first-business-case-study.Customer.order\_items` order\_items

ON orders.order\_id = order\_items.order\_id

LEFT JOIN `first-business-case-study.Customer.Customer\_info` customer

ON orders.customer\_id = customer.customer\_id

group by customer.customer\_state),

max\_time\_delivery as

(SELECT customer\_state,time\_to\_delivery,max\_time\_delivery FROM State\_time\_to\_delivery

WHERE max\_time\_delivery<=5),

min\_time\_delivery as

(SELECT customer\_state,time\_to\_delivery,min\_time\_delivery FROM State\_time\_to\_delivery

WHERE min\_time\_delivery<=5)

SELECT max\_time\_delivery.customer\_state as top\_five\_state,

max\_time\_delivery.time\_to\_delivery as top\_delivery\_time,

min\_time\_delivery.customer\_state as lower\_five\_state,

min\_time\_delivery.time\_to\_delivery as lowerest\_delivery\_time

FROM max\_time\_delivery

INNER JOIN min\_time\_delivery

ON max\_time\_delivery.max\_time\_delivery = min\_time\_delivery.min\_time\_delivery

order by max\_time\_delivery.max\_time\_delivery;

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

WITH fast\_delivery as

(SELECT distinct customer.customer\_state,

TIMESTAMP\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,HOUR) as time\_to\_delivery,

dense\_rank() over(partition by customer.customer\_state order by TIMESTAMP\_DIFF(orders.order\_delivered\_customer\_date, orders.order\_purchase\_timestamp,HOUR)) as d\_rnk

FROM `first-business-case-study.Customer.orders` orders

LEFT JOIN `first-business-case-study.Customer.order\_items` order\_items

ON orders.order\_id = order\_items.order\_id

LEFT JOIN `first-business-case-study.Customer.Customer\_info` customer

ON orders.customer\_id = customer.customer\_id

where TIMESTAMP\_DIFF(orders.order\_estimated\_delivery\_date, orders.order\_delivered\_customer\_date, HOUR)>0)

SELECT customer\_state, time\_to\_delivery FROM fast\_delivery where d\_rnk=1 order by time\_to\_delivery limit 5;

6.1. Month over Month count of orders for different payment types

SELECT count(orders.order\_id) as cnt\_order\_per\_month, EXTRACT(MONTH from orders.order\_purchase\_timestamp) as month\_order, payments.payment\_type

FROM `first-business-case-study.Customer.orders` orders

LEFT JOIN `first-business-case-study.Customer.payments` payments

ON orders.order\_id = payments.order\_id

group by EXTRACT(MONTH from orders.order\_purchase\_timestamp),payments.payment\_type;

6.2. Count of orders based on the no. of payment installments

SELECT payment\_installments, count(distinct order\_id) as cnt\_order FROM `first-business-case-study.Customer.payments`

group by payment\_installments

order by payment\_installments;