***1.2.*** select \* from

**PROJECT**

business.public.orders;

select

min(order\_purchase\_timestamp) as earliest\_order,

max(order\_purchase\_timestamp) as latest\_order,

datediff(hour, min(order\_purchase\_timestamp), max(order\_purchase\_timestamp)) as range

from

orders;

***1.3.*** select

min(order\_purchase\_timestamp) as earliest\_order,

max(order\_purchase\_timestamp) as latest\_order,

datediff(hour, min(order\_purchase\_timestamp), max(order\_purchase\_timestamp)) as range

from

orders;

select

c.customer\_city as city,

c.customer\_state as state,

count(distinct c.customer\_id) as numcustomer

from

customer c

join

orders o

on

c.customer\_id = o.customer\_id

where

o.order\_purchase\_timestamp between '2017-01-01' and '2018-06-30'

group by

c.customer\_city,

c.customer\_state

order by

numcustomer desc;

***2.1***-select

extract(year from order\_purchase\_timestamp) as order\_year,

count(order\_id) as num\_orders

from

orders

group by

order\_year

order by

order\_year;

***2.2***-select

extract(month from order\_purchase\_timestamp) as order\_month,

count(order\_id) as num\_orders

from

orders

group by

order\_month

order by

order\_month;

***2.3***-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'

else 'Unknown'

end as order\_time\_category,

count(order\_id) as num\_orders

from

orders

group by

order\_time\_category

order by

min(extract(hour from order\_purchase\_timestamp));

***3.1-***select

c.customer\_state,

date\_trunc('month', o.order\_purchase\_timestamp) as order\_month,

count(o.order\_id) as order\_count

from

orders o

join

customer c on o.customer\_id = c.customer\_id

group by

c.customer\_state,

date\_trunc ('month', o.order\_purchase\_timestamp)

order by

c.customer\_state,

order\_month;

***3.2***-select

customer\_state,

count(customer\_id) as customer\_count

from

customer

group by

customer\_state

order by

customer\_state;

***4.1***-with payments\_2017 as (

select

sum(p.payment\_value) as total\_payment\_value\_2017

from

payment p

join

orders o on p.order\_id = o.order\_id

where

o.order\_purchase\_timestamp between '2017-01-01' and '2017-08-31'

),

payments\_2018 as (

select

sum(p.payment\_value) as total\_payment\_value\_2018

from

payment p

join

orders o on p.order\_id = o.order\_id

where

o.order\_purchase\_timestamp between '2018-01-01' and '2018-08-31'

)

select

total\_payment\_value\_2017,

total\_payment\_value\_2018,

round(((total\_payment\_value\_2018 - total\_payment\_value\_2017) / total\_payment\_value\_2017) \* 100, 2) as percent\_increase

from

payments\_2017, payments\_2018;

***4.2-*** select c.customer\_state,

sum(oi.price) as total\_order\_price,

round(avg(oi.price), 2) as average\_order\_price

from

orders o

join

customer c on o.customer\_id = c.customer\_id

join

order\_items oi on o.order\_id = oi.order\_id

group by

c.customer\_state

order by

total\_order\_price desc;

***4.3-***select

c.customer\_state

sum(oi.freight\_value) as total\_freight\_value,

round(avg(oi.freight\_value), 2) as average\_freight\_value

from

orders o

join

customer c on o.customer\_id = c.customer\_id

join

order\_items oi on o.order\_id = oi.order\_id

group by

c.customer\_state

order by

total\_freight\_value desc;

***5.1***- select

customer\_state,

avg(datediff(day, o.order\_purchase\_timestamp::timestamp, o.order\_delivered\_customer\_date::timestamp)) as avg\_delivery\_time\_days

from

orders o

join

customer c on o.customer\_id = c.customer\_id

where

o.order\_status = 'delivered'

group by

customer\_state

order by

avg\_delivery\_time\_days desc

limit 5;

select

customer\_state,

avg(datediff(day, o.order\_purchase\_timestamp::timestamp, o.order\_delivered\_customer\_date::timestamp)) as avg\_delivery\_time\_days

from

orders o

join

customer c on o.customer\_id = c.customer\_id

where

o.order\_status = 'delivered'

group by

customer\_state

order by

avg\_delivery\_time\_days asc

limit 5;

***5.2***-with avg\_freight\_values as(

Select

c.customer\_state,

avg(oi.freight\_value) as avg\_freight\_value,

row\_number() over (order by avg(oi.freight\_value) desc) as highest\_rank,

row\_number() over (order by avg (oi.freight\_value) asc) as lowest\_rank

from

order\_items oi

join

orders o on oi.order\_id = o.order\_id

join

customer c on o.customer\_id = c.customer\_id

where

oi.freight\_value is not null

group by

c.customer\_state

)

select

customer\_state,

avg\_freight\_value,

case when highest\_rank <= 5 THEN 'Highest' else 'Lowest' end as freight\_rank

from

avg\_freight\_values

where

highest\_rank <= 5 or lowest\_rank <= 5

order by

case when freight\_rank = 'Highest' THEN avg\_freight\_value end desc,

case when freight\_rank = 'Lowest' THEN avg\_freight\_value end asc;

***5.3***-with avg\_delivery\_times as (

select

c.customer\_state,

avg(datediff(day, o.order\_purchase\_timestamp, o.order\_delivered\_customer\_date)) as avg\_delivery\_time,

row\_number() over (order by avg(datediff(day, o.order\_purchase\_timestamp, o.order\_delivered\_customer\_date)) desc) as highest\_rank,

row\_number () over (order by avg(datediff (day, o.order\_purchase\_timestamp, o.order\_delivered\_customer\_date)) asc) as lowest\_rank

from

orders o

join

customer c on o.customer\_id = c.customer\_id

where

o.order\_delivered\_customer\_date is not null

group by

c.customer\_state

)

select

customer\_state,

avg\_delivery\_time,

case when highest\_rank <= 5 THEN 'Highest' else 'Lowest' end AS delivery\_rank

from

avg\_delivery\_times

where

highest\_rank <= 5 or lowest\_rank <= 5

order by

case when delivery\_rank = 'Highest' THEN avg\_delivery\_time end desc,

case when delivery\_rank = 'Lowest' THEN avg\_delivery\_time end asc;

***5.4-***with avg\_dates as (

select

c.customer\_state,

avg(datediff(day, o.order\_purchase\_timestamp, o.order\_delivered\_customer\_date)) as avg\_actual\_delivery\_days,

avg(datediff(day, o.order\_purchase\_timestamp, o.order\_estimated\_delivery\_date)) as avg\_estimated\_delivery\_days

from

orders o

join

customer c on o.customer\_id = c.customer\_id

where

o.order\_delivered\_customer\_date is not null

and o.order\_estimated\_delivery\_date is not null

group by

c.customer\_state

)

select

customer\_state,

avg\_actual\_delivery\_days,

avg\_estimated\_delivery\_days,

avg\_estimated\_delivery\_days - avg\_actual\_delivery\_days as avg\_delivery\_difference

from

avg\_dates

order by

avg\_delivery\_difference desc

limit 5;

***6.1-***select

extract(year from o.order\_purchase\_timestamp) as order\_year,

extract (month from o.order\_purchase\_timestamp) as order\_month,

p.payment\_type,

count (o.order\_id) as num\_orders

from

orders o

join

Payment p on o.order\_id = p.order\_id

Group by

order\_year, order\_month, p.payment\_type

order by

order\_year, order\_month, num\_orders desc;

***6.2***-select

p.payment\_installments,

count(o.order\_id) as num\_orders

from

payment p

join

orders o on p.order\_id = o.order\_id

group by

p.payment\_installments

order by

p.payment\_installments;