/\*

All the challenges can be found on https://8weeksqlchallenge.com/case-study-3/

1.How many customers has Foodie-Fi ever had?

2.What is the monthly distribution of trial plan start\_date values for our dataset - use the start of the month as the group by value

3.What plan start\_date values occur after the year 2020 for our dataset? Show the breakdown by count of events for each plan\_name

4.What is the customer count and percentage of customers who have churned rounded to 1 decimal place?

5.How many customers have churned straight after their initial free trial - what percentage is this rounded to the nearest whole number?

6.What is the number and percentage of customer plans after their initial free trial?

7.What is the customer count and percentage breakdown of all 5 plan\_name values at 2020-12-31?

8.How many customers have upgraded to an annual plan in 2020?

9.How many days on average does it take for a customer to an annual plan from the day they join Foodie-Fi?

10.Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)

11.How many customers downgraded from a pro monthly to a basic monthly plan in 2020?

\*/

----------------------------------------------

--1.How many customers has Foodie-Fi ever had?

----------------------------------------------

select count(distinct customer\_id)

from subscriptions;

---------------------------------------------------------------------------------------------------------------------------------------

--2.What is the monthly distribution of trial plan start\_date values for our dataset - use the start of the month as the group by value

---------------------------------------------------------------------------------------------------------------------------------------

select date\_trunc('month',start\_date) as month,

count(date\_trunc('month',start\_date)) as count

from subscriptions

where plan\_id = 0

group by month

order by month;

------------------------------------------------------------------------------------------------------------------------------------

--3. What plan start\_date values occur after the year 2020 for our dataset? Show the breakdown by count of events for each plan\_name

------------------------------------------------------------------------------------------------------------------------------------

select

count(subscriptions.plan\_id),

max(plan\_name),

left(start\_date, 4) as year

from subscriptions

join plans

on subscriptions.plan\_id = plans.plan\_id

where left(start\_date, 4) > 2020

group by subscriptions.plan\_id, year;

-------------------------------------------------------------------------------------------------------

--4.What is the customer count and percentage of customers who have churned rounded to 1 decimal place?

-------------------------------------------------------------------------------------------------------

select count(customer\_id) as churned\_customers

,round(max(percent\_of\_churned\_customers\*100),1) as percent\_of\_churned\_customers

from subscriptions

join(

select max(

select count(customer\_id)

from subscriptions

where plan\_id = 4

)/

max(

select count(distinct customer\_id)

from subscriptions

) as percent\_of\_churned\_customers

from subscriptions

)

where plan\_id = 4;

-------------------------------------------------------------------------------------------------------------------------------------------

--5. How many customers have churned straight after their initial free trial - what percentage is this rounded to the nearest whole number?

-------------------------------------------------------------------------------------------------------------------------------------------

select count(churn\_date) as left\_after\_trial,

max(all\_customers) as all\_customers,

round((left\_after\_trial/max(all\_customers))\*100,2) as percent\_left\_after\_trial

from(

select subscriptions.customer\_id,

temp1.start\_date,

subscriptions.start\_date as churn\_date,

datediff(days, temp1.start\_date,churn\_date) as days

from subscriptions

join(

select customer\_id,

max(plan\_id),

min(start\_date) as start\_date

from subscriptions

where plan\_id != 4

group by customer\_id

) as temp1

on subscriptions.customer\_id = temp1.customer\_id

where subscriptions.plan\_id = 4

and days <=7

)

join(

select count(distinct customer\_id) as all\_customers

from subscriptions

) as temp1;

-- wenn "as temp1 nicht hinzugefügt wird, erscheint error message: "SQL compilation error: duplicate alias 'values'"

----------------------------------------------------------------------------------------

--6. What is the number and percentage of customer plans after their initial free trial?

----------------------------------------------------------------------------------------

select count(distinct customer\_id) as all\_customers,

max(plan) as plan,

round(max(plan)/all\_customers\*100, 2 )as percent\_went\_on\_with\_plan

from subscriptions

join(

select count(distinct customer\_id) as plan

from subscriptions

where plan\_id != 4

and plan\_id != 0

);

------------------------------------------------------------------------------------------------

--7.What is the customer count and percentage breakdown of all 5 plan\_name values at 2020-12-31?

------------------------------------------------------------------------------------------------

select count(customer\_id) as cus\_per\_plan,

round(cus\_per\_plan/max(all\_customers)\*100,2) as percent\_of\_plans,

plan\_id

from subscriptions

join(

select count(distinct customer\_id) as all\_customers

from subscriptions

)

where start\_date < to\_date('2020-12-31')

group by plan\_id;

---------------------------------------------------------------

--8.How many customers have upgraded to an annual plan in 2020?

---------------------------------------------------------------

select count(distinct subscriptions.customer\_id) as cus\_who\_upgraded\_to\_3

from subscriptions

join(

select customer\_id

from subscriptions

where plan\_id = 3

and start\_date like '%2020%'

group by customer\_id

) as prior\_plans

on subscriptions.customer\_id = prior\_plans.customer\_id

where plan\_id in (0,1,2);

------------------------------------------------------------------------------------------------------------

--9.How many days on average does it take for a customer to an annual plan from the day they join Foodie-Fi?

------------------------------------------------------------------------------------------------------------

select max(avg\_to\_upgrade) as avg\_days\_to\_upgrade

from(

select

--subscriptions.customer\_id

--,min(to\_date(subscriptions.start\_date)) as start\_date

--,max(update\_date) as update\_date

--,

avg(max(datediff(day, to\_date(start\_date), to\_date(update\_date)))) over() as avg\_to\_upgrade

--,max(datediff(day, to\_date(start\_date), to\_date(update\_date)))

from subscriptions

join(

select customer\_id,

max(plan\_id) as plan\_id\_3,

max(to\_date(start\_date)) as update\_date

from subscriptions

where plan\_id = 3

group by customer\_id

) as prior\_plans

on subscriptions.customer\_id = prior\_plans.customer\_id

where plan\_id = 0

group by subscriptions.customer\_id

);

------------------------------------------------------------------------------------------------------

--10.Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)

------------------------------------------------------------------------------------------------------

select count(customer) as num\_of\_cus,

case

when days <= 30 then '0-30'

when days > 30 and days < 60 then '30-60'

else '>60'

end as period

from(

--

select subscriptions.customer\_id as customer

,min(to\_date(subscriptions.start\_date)) as start\_date

,max(update\_date) as update\_date

,avg(max(datediff(day, to\_date(start\_date), to\_date(update\_date)))) over() as avg\_to\_upgrade

,max(datediff(day, to\_date(start\_date), to\_date(update\_date))) as days

/\* geht nicht, da dann immer nur das maximum für alle zeilen genommen wird

max(

case when subscriptions.customer\_id = 2 then '0-30'

when days > 30 and days < 60 then '30-60'

else '>60'

end

) over() as periods

\*/

from subscriptions

join(

select customer\_id,

max(plan\_id) as plan\_id\_3,

max(to\_date(start\_date)) as update\_date

from subscriptions

where plan\_id = 3

group by customer\_id

) as prior\_plans

on subscriptions.customer\_id = prior\_plans.customer\_id

where plan\_id = 0

group by subscriptions.customer\_id

)

group by period;

--------------------------------------------------------------------------------------

--11.How many customers downgraded from a pro monthly to a basic monthly plan in 2020?

--------------------------------------------------------------------------------------

select

--subscriptions.customer\_id

--,pro\_month\_upgrade

--,subscriptions.start\_date

--,

count(case

when to\_date(pro\_month\_upgrade) >= to\_date(start\_date)

then 'upgrade'

else 'downgrade'

end) as up\_or\_down

from subscriptions

join(

select customer\_id,

start\_date as pro\_month\_upgrade

from subscriptions

where plan\_id = 2 --pro monthly

and start\_date like '%2020%'

) as pro\_monthly

on subscriptions.customer\_id = pro\_monthly.customer\_id

where plan\_id = 1

and start\_date like '%2020%'

group by (to\_date(pro\_month\_upgrade) > to\_date(start\_date)); --basic monthly

----------------------------------------------------------------

--end of challenges---------------------------------------------

----------------------------------------------------------------

--extra: mit over() können zwei aggregations verschachtelt werden

select sum(count(date\_trunc('month', start\_date))) over() as total

from subscriptions

where plan\_id=0

group by date\_trunc('month', start\_date);

--check if customers, that have a startdate in 2021 have also a startdate in 2020

select customer\_id, start\_date

from subscriptions

where customer\_id in (

select customer\_id

from subscriptions

where start\_date like '%2021%'

)

and start\_date like '%2020%'

order by customer\_id;