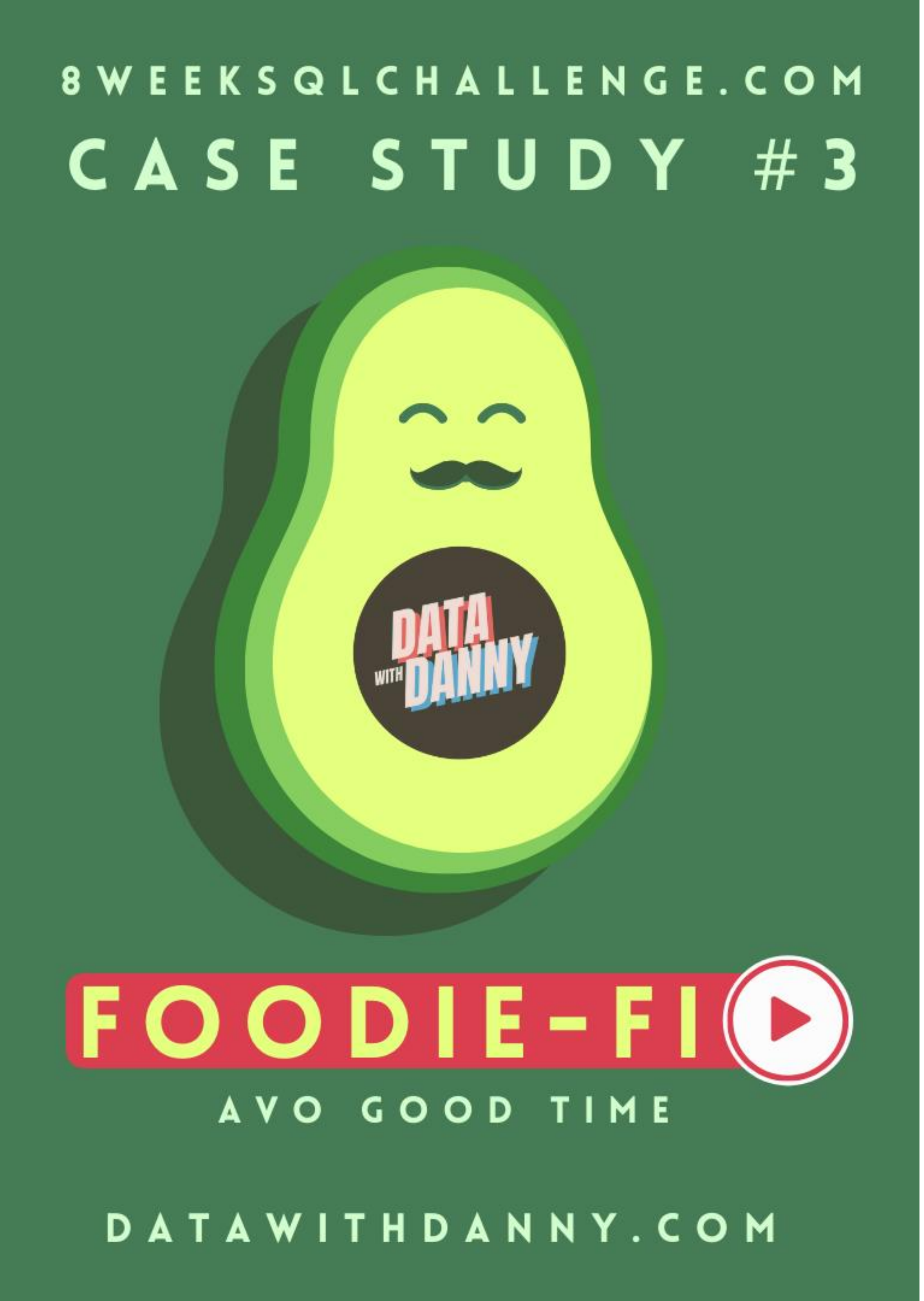
Danny Ma Case Study 3: Foodie-Fi

By : Shikhar Chopra

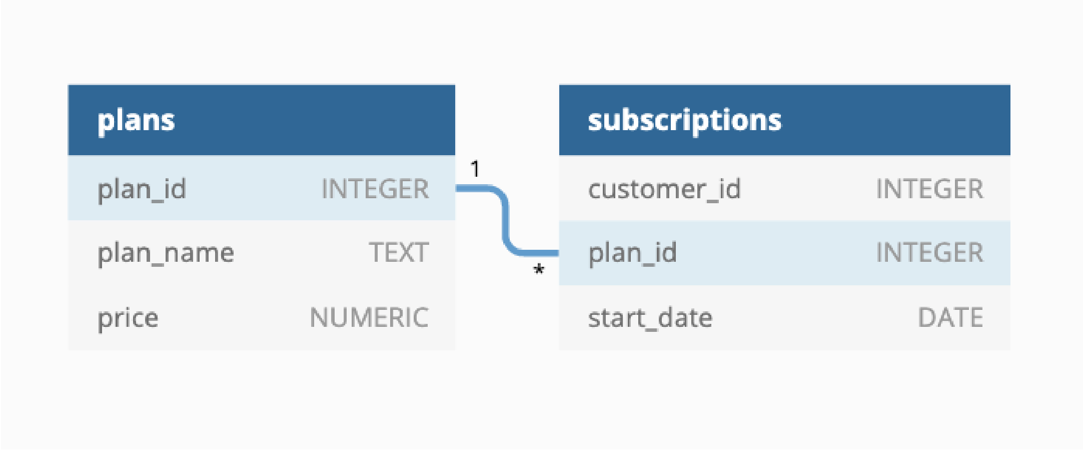
Tool used – Microsoft SQL Server



Introduction

- Subscription based businesses are super popular and Danny realised that there was a large gap in the market - he wanted to create a new streaming service that only had food related content - something like Netflix but with only cooking shows!
- Danny finds a few smart friends to launch his new startup Foodie-Fi in 2020 and started selling monthly and annual subscriptions, giving their customers unlimited on-demand access to exclusive food videos from around the world!
- Danny created Foodie-Fi with a data driven mindset and wanted to ensure all future investment decisions and new features were decided using data. This case study focuses on using subscription style digital data to answer important business questions.

Available Data



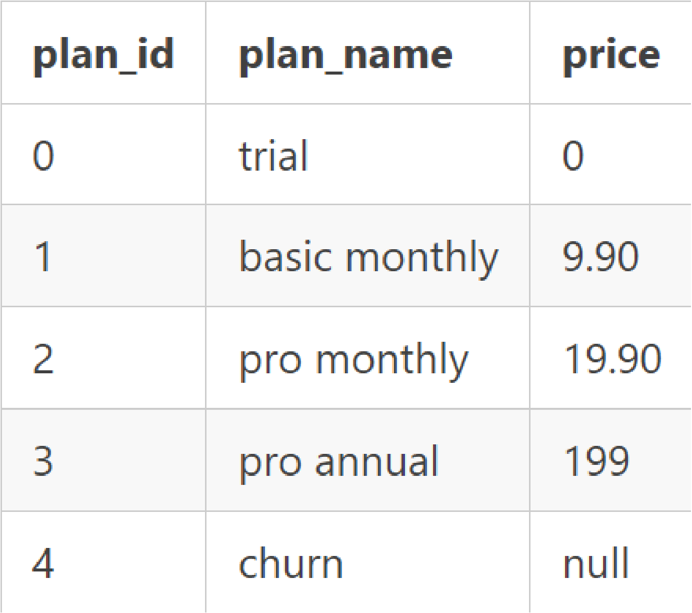
plans	
plan_id	INTEGER
plan_name	TEXT
price	NUMERIC

subscriptions	
customer_id	INTEGER
plan_id	INTEGER
start_date	DATE

Danny has shared the data design for Foodie-Fi and also short descriptions on each of the database tables - our case study focuses on only 2 tables but there will be a challenge to create a new table for the Foodie-Fi team.

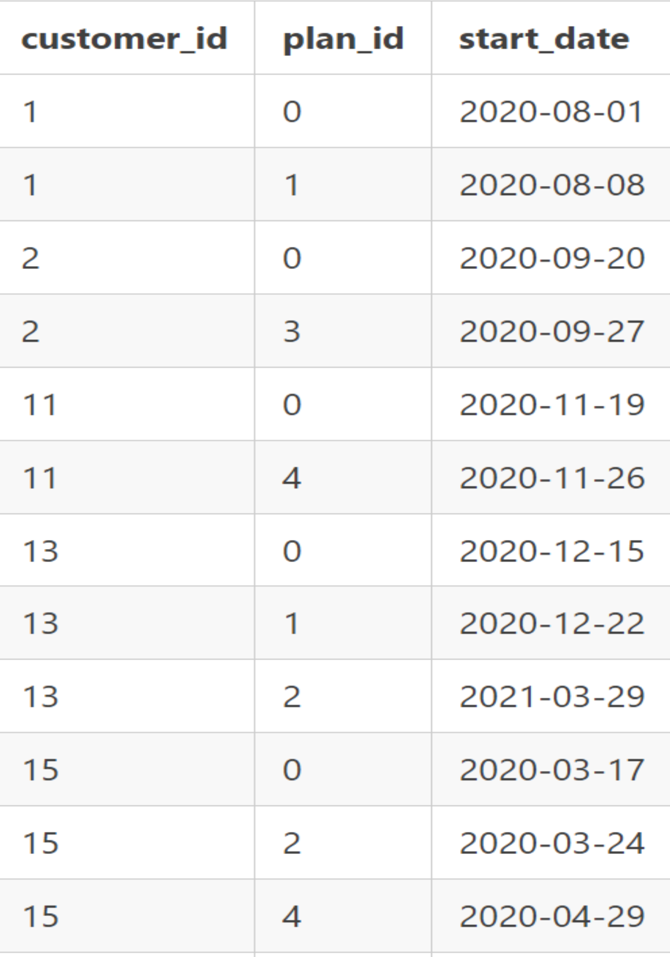
Tables

Table 1: plans



plan_id	plan_name	price
0	trial	0
1	basic monthly	9.90
2	pro monthly	19.90
3	pro annual	199
4	churn	null

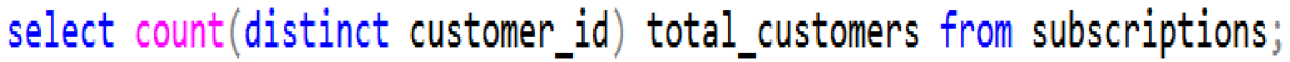
Table 2: subscriptions



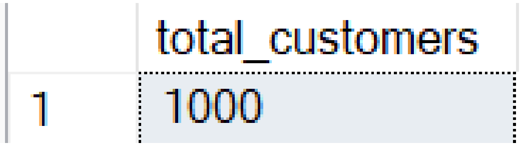
customer_id	plan_id	start_date
1	0	2020-08-01
1	1	2020-08-08
2	0	2020-09-20
2	3	2020-09-27
11	0	2020-11-19
11	4	2020-11-26
13	0	2020-12-15
13	1	2020-12-22
13	2	2021-03-29
15	0	2020-03-17
15	2	2020-03-24
15	4	2020-04-29

Case Study Questions

Q1. How many customers has Foodie-Fi ever had?

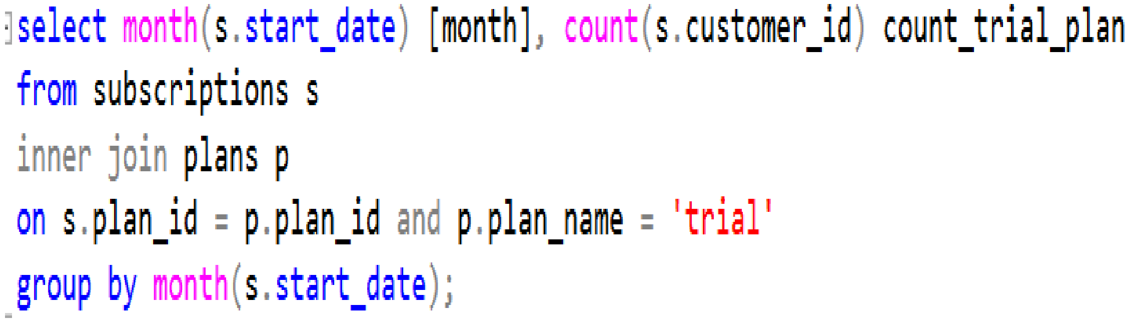


```
select count(distinct customer_id) total_customers from subscriptions;
```

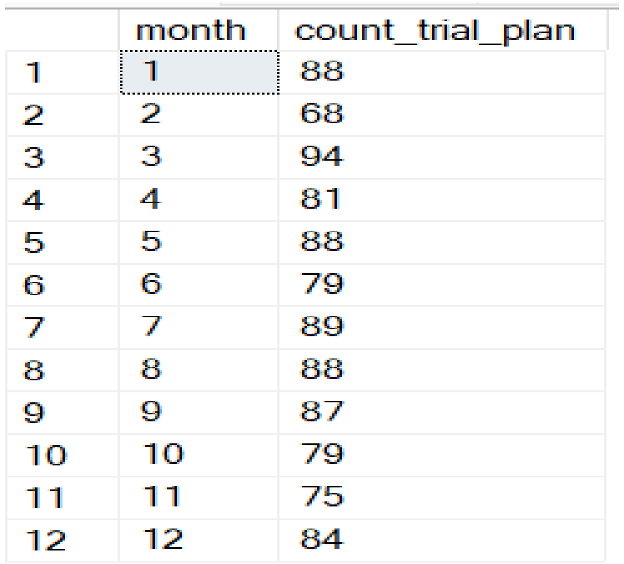


	total_customers
1	1000

Q2. What is the monthly distribution of trial plan start_date values for our dataset

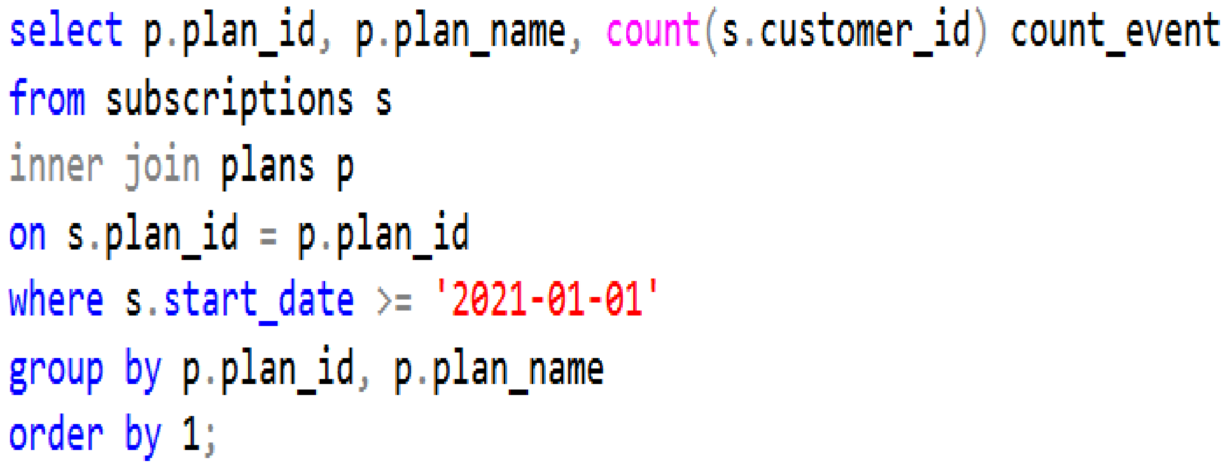


```
select month(s.start_date) [month], count(s.customer_id) count_trial_plan
from subscriptions s
inner join plans p
on s.plan_id = p.plan_id and p.plan_name = 'trial'
group by month(s.start_date);
```

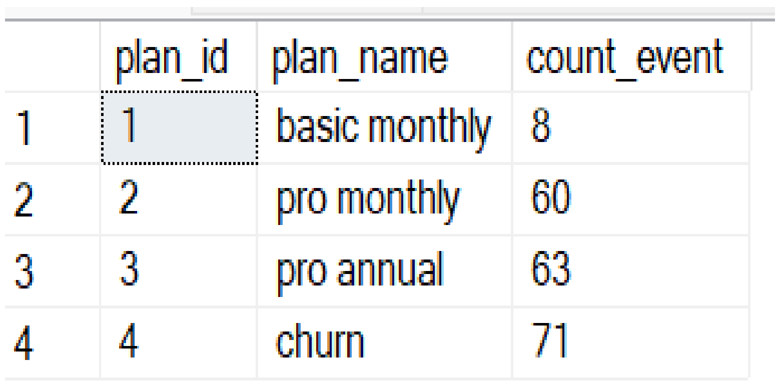


	month	count_trial_plan
1	1	88
2	2	68
3	3	94
4	4	81
5	5	88
6	6	79
7	7	89
8	8	88
9	9	87
10	10	79
11	11	75
12	12	84

Q3. What plan start_date values occur after the year 2020 for our dataset? Show the breakdown by count of events for each plan_name

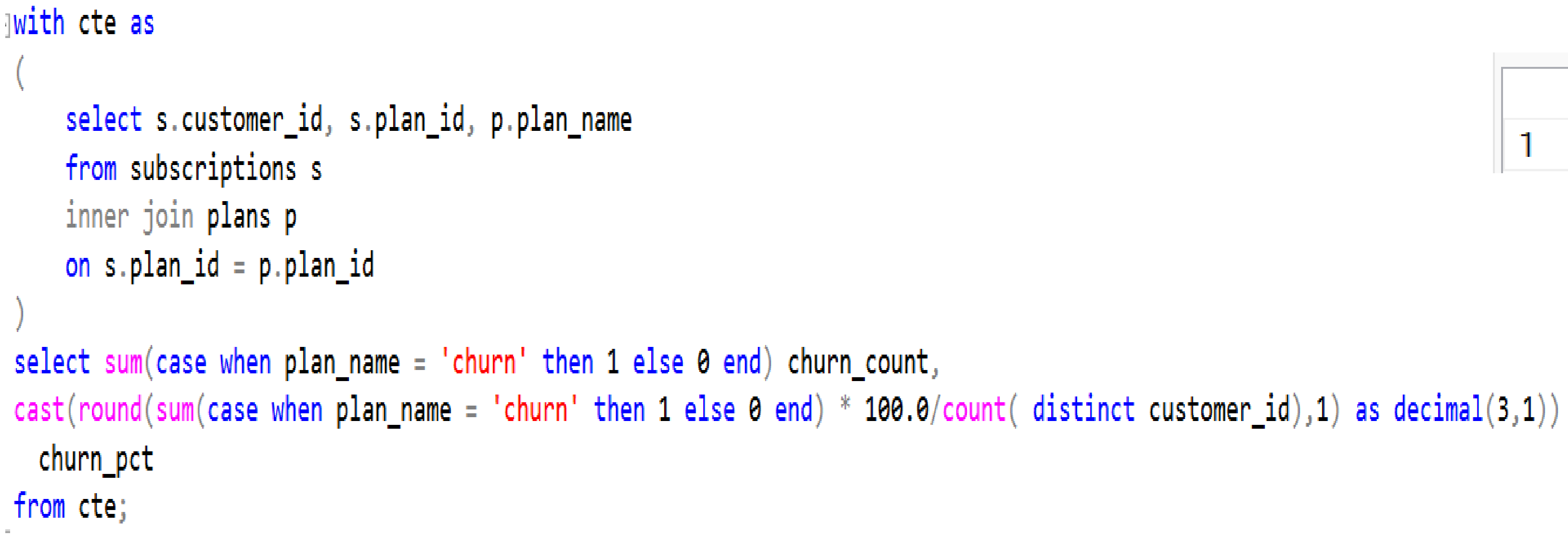


```sql
select p.plan_id, p.plan_name, count(s.customer_id) count_event
from subscriptions s
inner join plans p
on s.plan_id = p.plan_id
where s.start_date >= '2021-01-01'
group by p.plan_id, p.plan_name
order by 1;
```

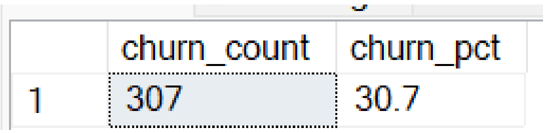


	plan_id	plan_name	count_event
1	1	basic monthly	8
2	2	pro monthly	60
3	3	pro annual	63
4	4	churn	71

Q4. What is the customer count and percentage of customers who have churned rounded to 1 decimal place?

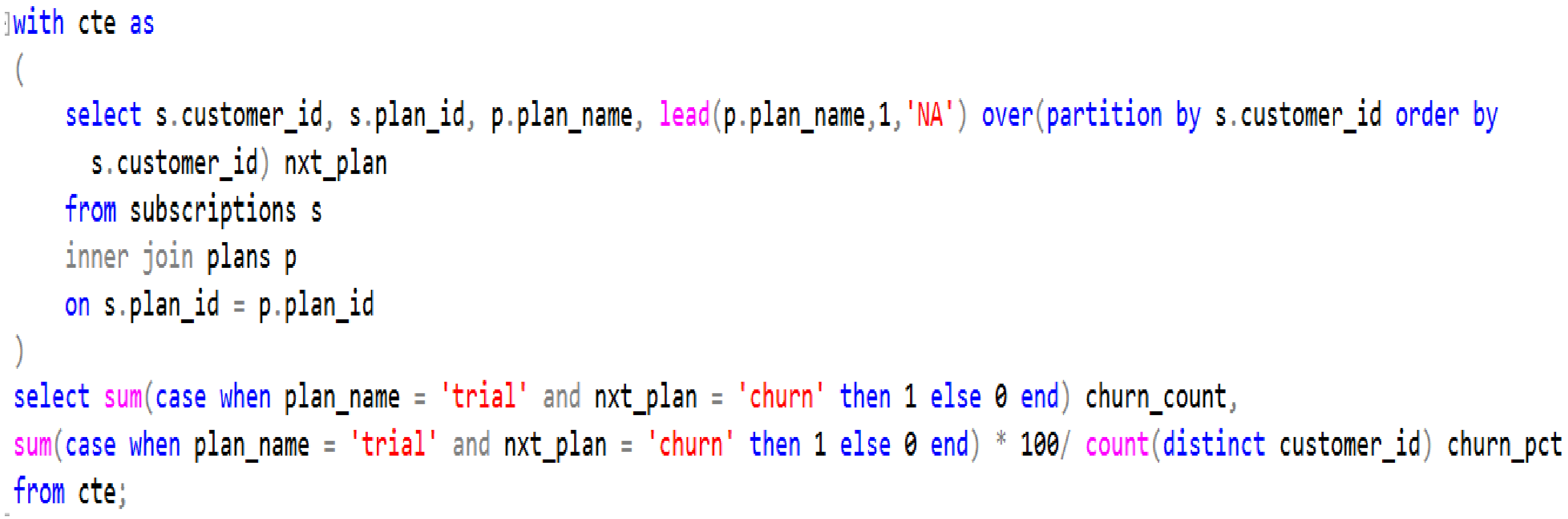


```
with cte as
(
    select s.customer_id, s.plan_id, p.plan_name
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
)
select sum(case when plan_name = 'churn' then 1 else 0 end) churn_count,
cast(round(sum(case when plan_name = 'churn' then 1 else 0 end) * 100.0/count( distinct customer_id),1) as decimal(3,1))
  churn_pct
from cte;
```

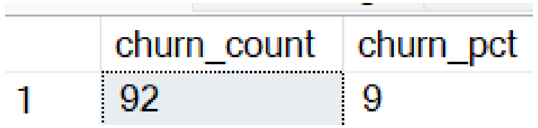


	churn_count	churn_pct
1	307	30.7

Q5. How many customers have churned straight after their initial free trial - what percentage is this rounded to the nearest whole number?

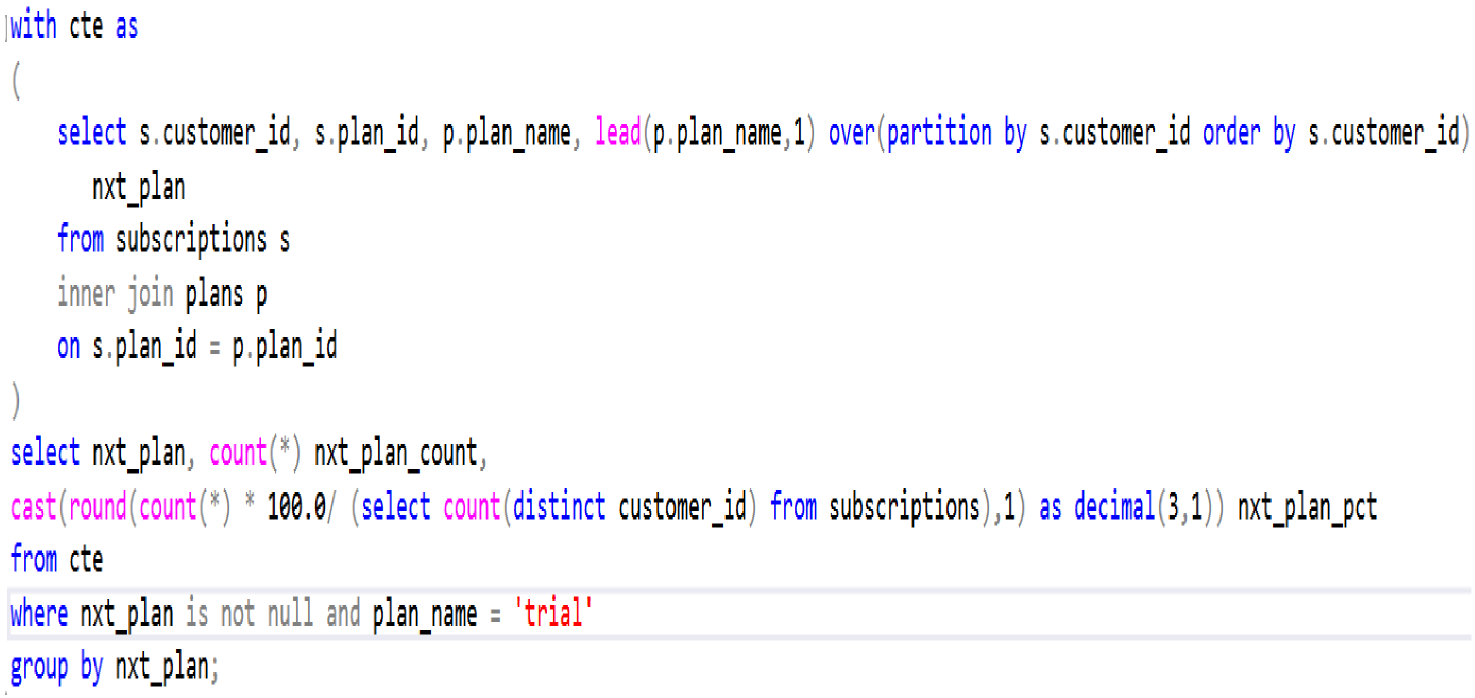


```
with cte as
(
    select s.customer_id, s.plan_id, p.plan_name, lead(p.plan_name,1,'NA') over(partition by s.customer_id order by
      s.customer_id) nxt_plan
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
)
select sum(case when plan_name = 'trial' and nxt_plan = 'churn' then 1 else 0 end) churn_count,
sum(case when plan_name = 'trial' and nxt_plan = 'churn' then 1 else 0 end) * 100/ count(distinct customer_id) churn_pct
from cte;
```

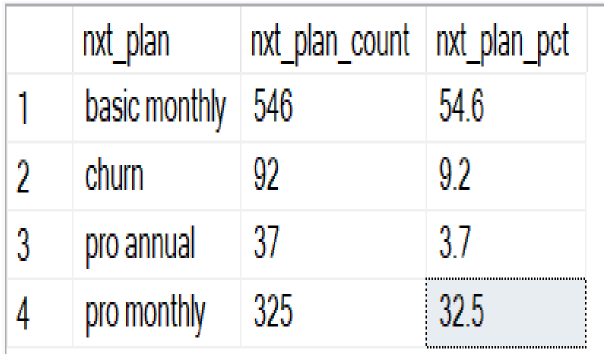


	churn_count	churn_pct
1	92	9

Q6. What is the number and percentage of customer plans after their initial free trial?

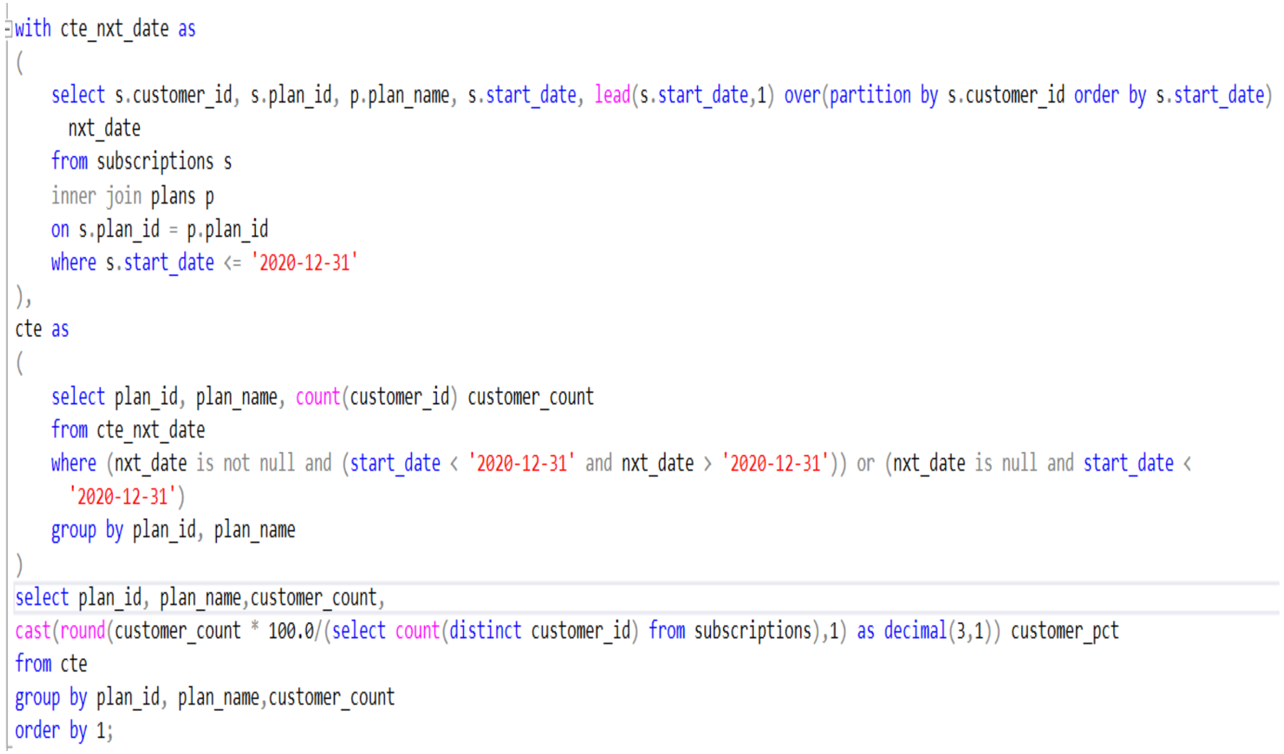


```
with cte as
(
    select s.customer_id, s.plan_id, p.plan_name, lead(p.plan_name,1) over(partition by s.customer_id order by s.customer_id)
      nxt_plan
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
)
select nxt_plan, count(*) nxt_plan_count,
cast(round(count(*) * 100.0/ (select count(distinct customer_id) from subscriptions),1) as decimal(3,1)) nxt_plan_pct
from cte
where nxt_plan is not null and plan_name = 'trial'
group by nxt_plan;
```

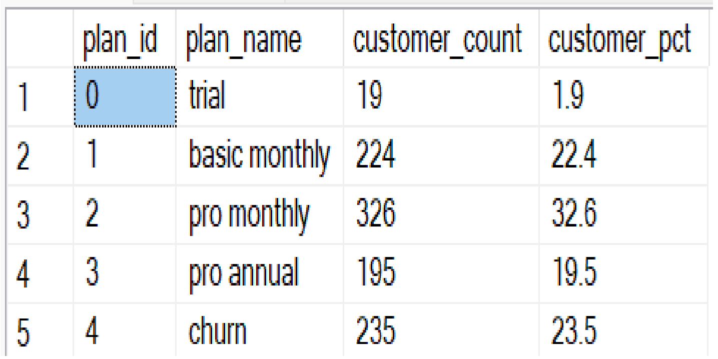


	nxt_plan	nxt_plan_count	nxt_plan_pct
1	basic monthly	546	54.6
2	churn	92	9.2
3	pro annual	37	3.7
4	pro monthly	325	32.5

Q7. What is the customer count and percentage breakdown of all 5 plan_name values at 2020-12-31?

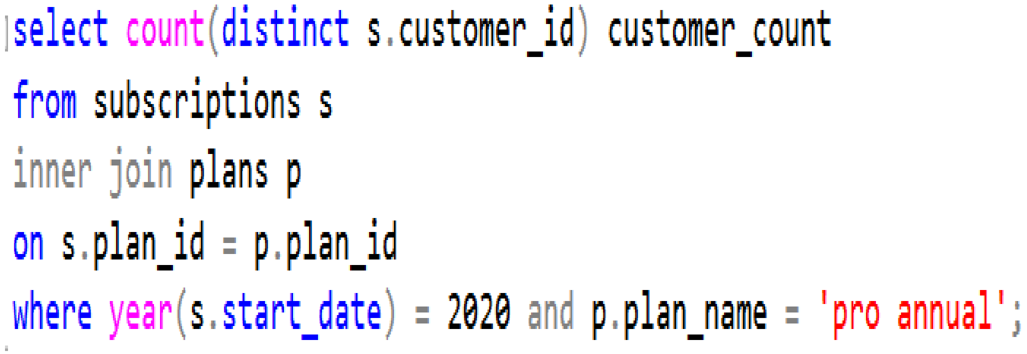


```
with cte_nxt_date as
(
    select s.customer_id, s.plan_id, p.plan_name, s.start_date, lead(s.start_date,1) over(partition by s.customer_id order by s.start_date)
      nxt_date
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
    where s.start_date <= '2020-12-31'
),
cte as
(
    select plan_id, plan_name, count(customer_id) customer_count
    from cte_nxt_date
    where (nxt_date is not null and (start_date < '2020-12-31' and nxt_date > '2020-12-31')) or (nxt_date is null and start_date <
      '2020-12-31')
    group by plan_id, plan_name
)
select plan_id, plan_name,customer_count,
cast(round(customer_count * 100.0/(select count(distinct customer_id) from subscriptions),1) as decimal(3,1)) customer_pct
from cte
group by plan_id, plan_name,customer_count
order by 1;
```

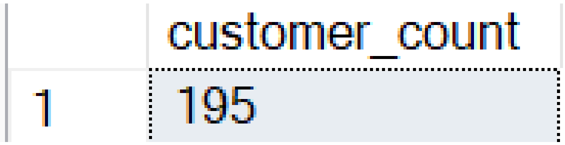


	plan_id	plan_name	customer_count	customer_pct
1	0	trial	19	1.9
2	1	basic monthly	224	22.4
3	2	pro monthly	326	32.6
4	3	pro annual	195	19.5
5	4	churn	235	23.5

Q8. How many customers have upgraded to an annual plan in 2020?

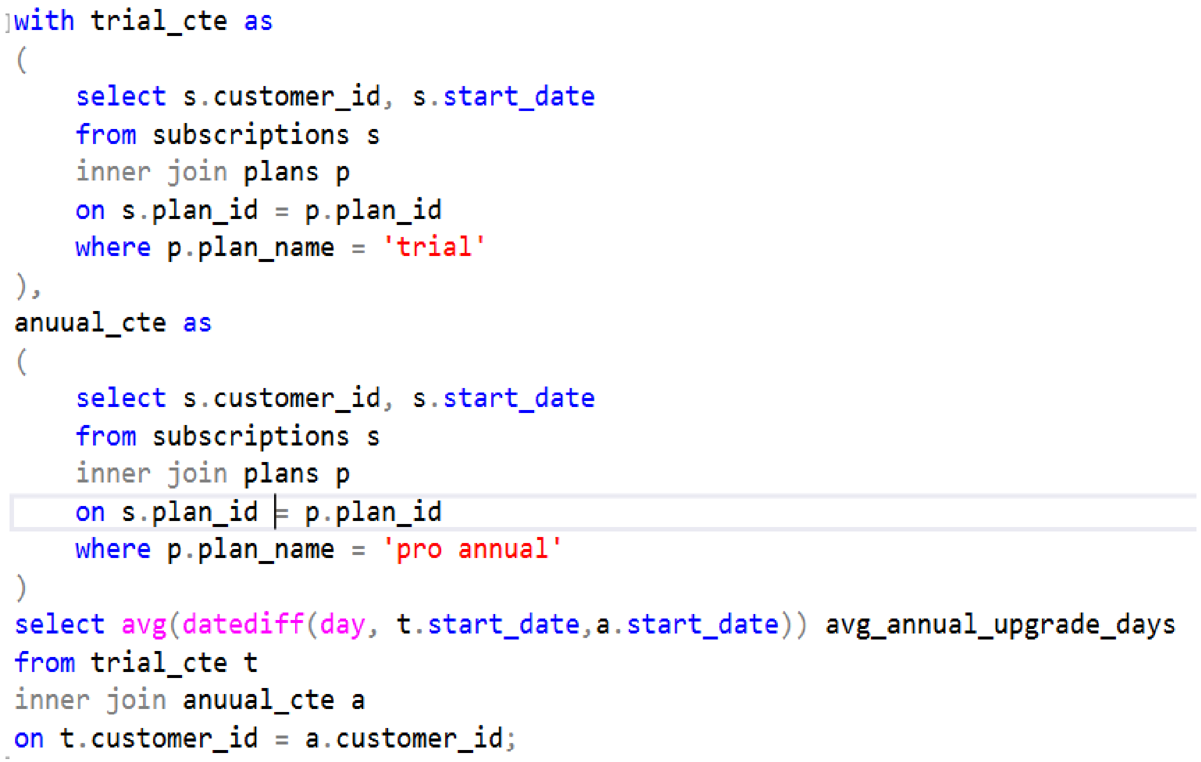


```
select count(distinct s.customer_id) customer_count
from subscriptions s
inner join plans p
on s.plan_id = p.plan_id
where year(s.start_date) = 2020 and p.plan_name = 'pro annual';
```

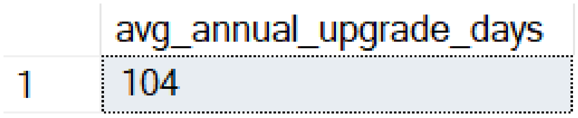


	customer_count
1	195

Q9. How many days on average does it take for a customer to an annual plan from the day they join Foodie-Fi?

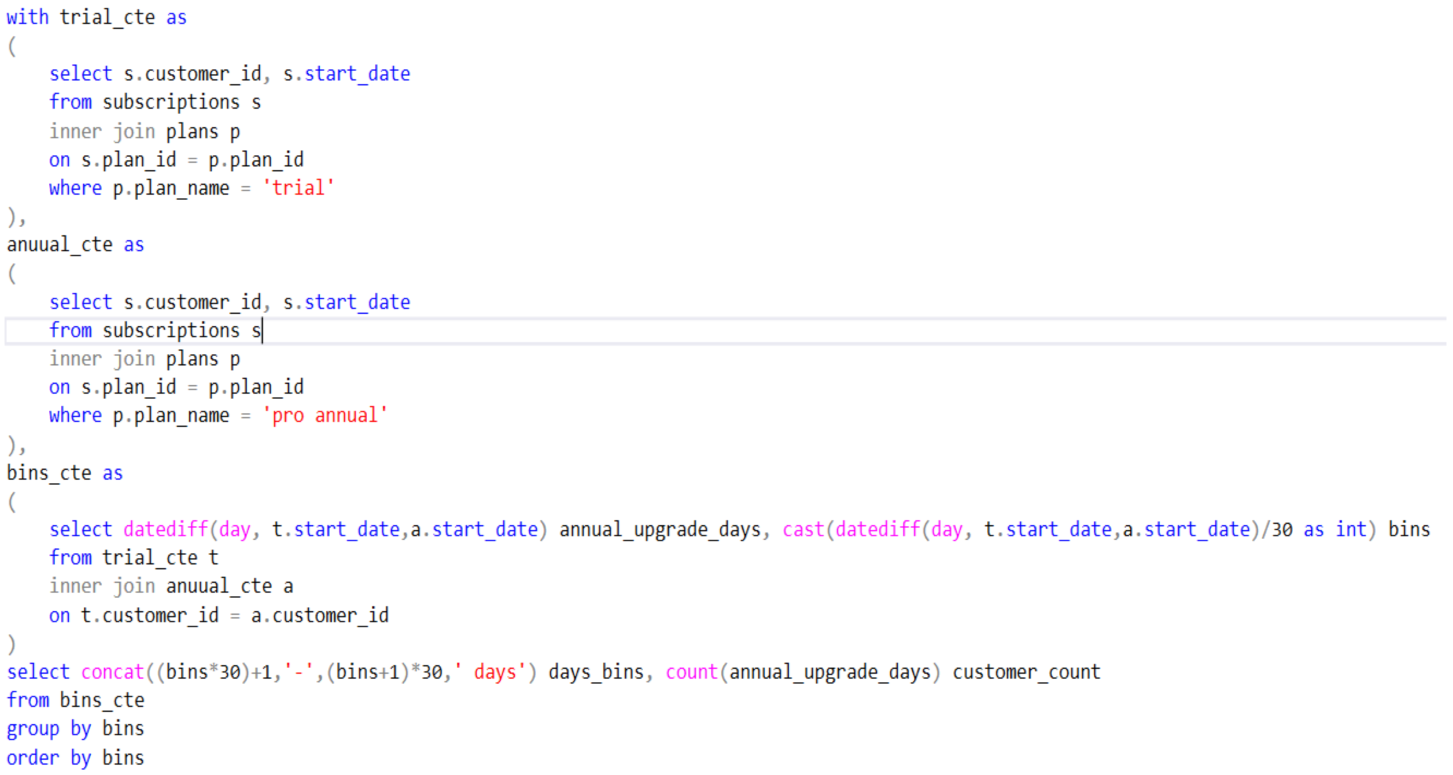


```
with trial_cte as
(
    select s.customer_id, s.start_date
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
    where p.plan_name = 'trial'
),
anuual_cte as
(
    select s.customer_id, s.start_date
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
    where p.plan_name = 'pro annual'
)
select avg(datediff(day, t.start_date,a.start_date)) avg_annual_upgrade_days
from trial_cte t
inner join anuual_cte a
on t.customer_id = a.customer_id;
```

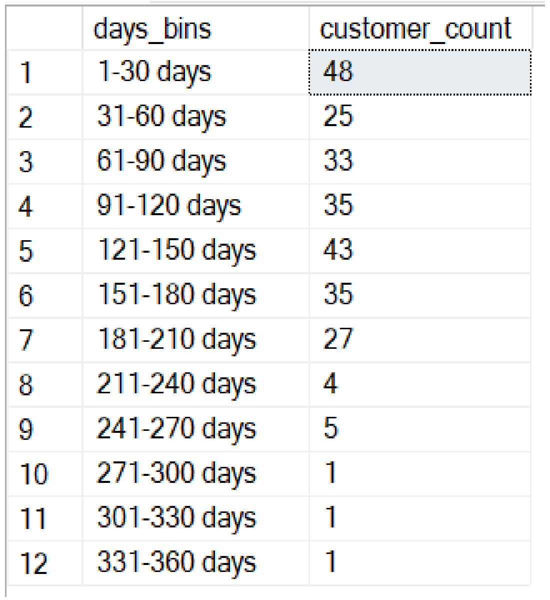


	avg_annual_upgrade_days
1	104

Q10. Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)?

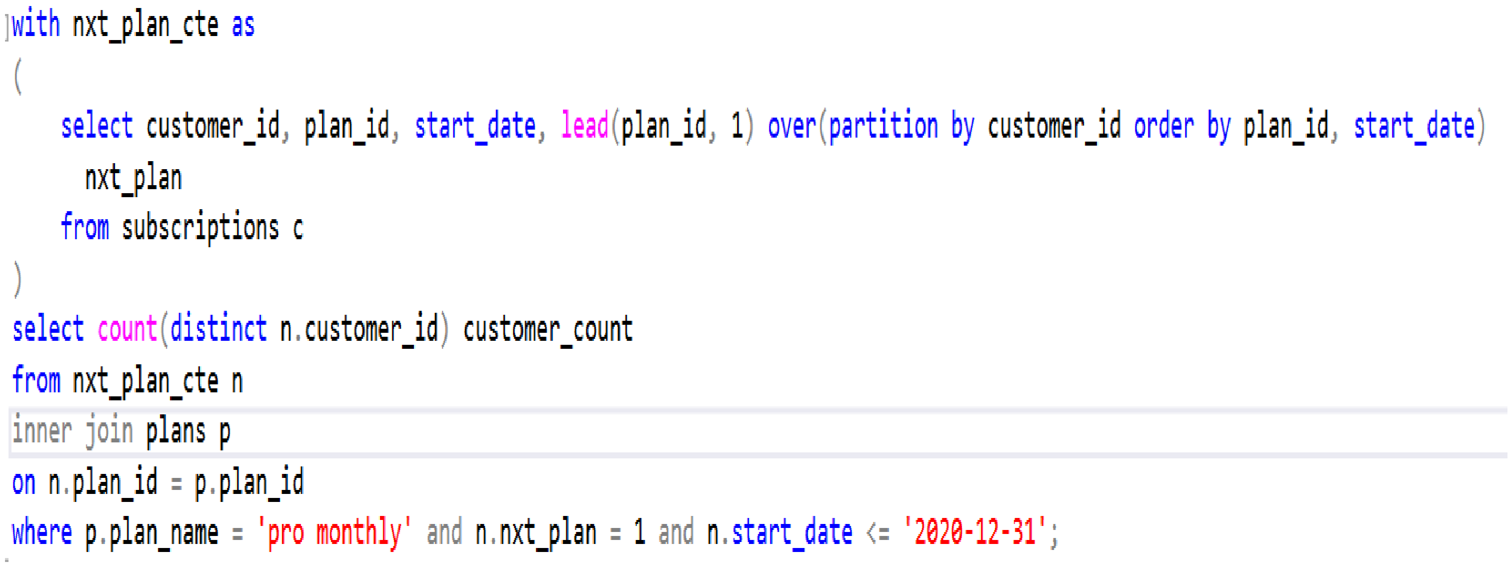


```
with trial_cte as
(
    select s.customer_id, s.start_date
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
    where p.plan_name = 'trial'
),
anuual_cte as
(
    select s.customer_id, s.start_date
    from subscriptions s
    inner join plans p
    on s.plan_id = p.plan_id
    where p.plan_name = 'pro annual'
),
bins_cte as
(
    select datediff(day, t.start_date,a.start_date) annual_upgrade_days, cast(datediff(day, t.start_date,a.start_date)/30 as int) bins
    from trial_cte t
    inner join anuual_cte a
    on t.customer_id = a.customer_id
)
select concat((bins*30)+1,'-',(bins+1)*30,' days') days_bins, count(annual_upgrade_days) customer_count
from bins_cte
group by bins
order by bins
```

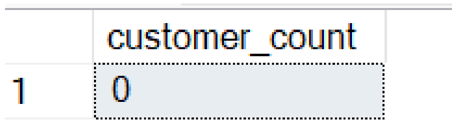


	days_bins	customer_count
1	1-30 days	48
2	31-60 days	25
3	61-90 days	33
4	91-120 days	35
5	121-150 days	43
6	151-180 days	35
7	181-210 days	27
8	211-240 days	4
9	241-270 days	5
10	271-300 days	1
11	301-330 days	1
12	331-360 days	1

Q11. How many customers downgraded from a pro monthly to a basic monthly plan in 2020?



```
with nxt_plan_cte as
(
    select customer_id, plan_id, start_date, lead(plan_id, 1) over(partition by customer_id order by plan_id, start_date)
      nxt_plan
    from subscriptions c
)
select count(distinct n.customer_id) customer_count
from nxt_plan_cte n
inner join plans p
on n.plan_id = p.plan_id
where p.plan_name = 'pro monthly' and n.nxt_plan = 1 and n.start_date <= '2020-12-31';
```



```
   customer_count
1  0
```