



- Aashik.N

Introduction

Danny 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!

- something like Netflix but with only cooking shows!

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

Available Data

- our case study focuses on only 2 tables
- All datasets exist within the foodie_fi database schema

Entity Relationship Diagram

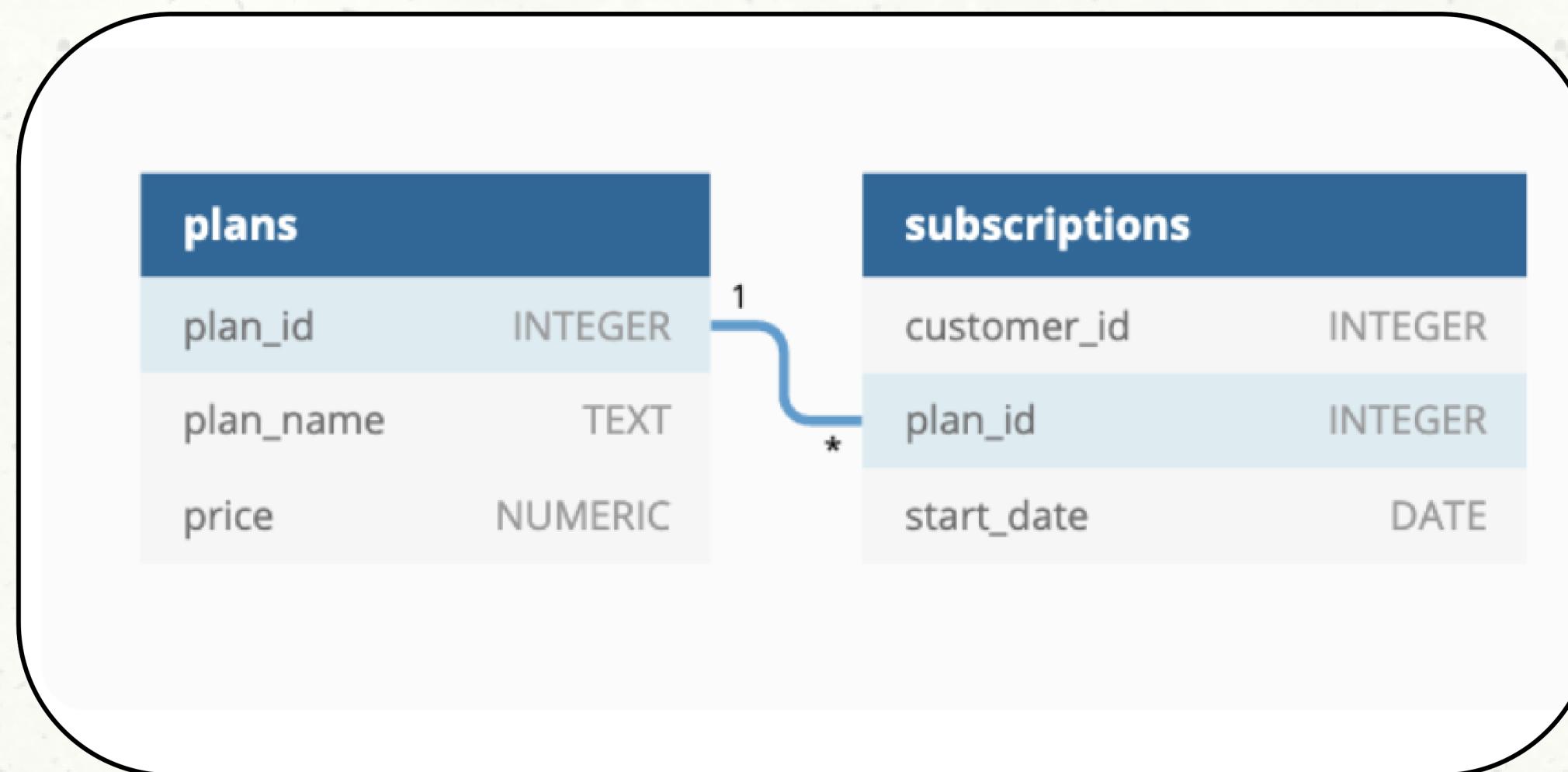


Table 1

Customers can sign up to an initial 7 day free trial will automatically continue with the pro monthly subscription plan unless they cancel, downgrade to basic or upgrade to an annual pro plan at any point during the trial.

- Basic plan monthly at \$9.90
- Pro plans start at \$19.90 a month
- \$199 for an pro annual subscription

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

Customer subscriptions show the exact date where their specific plan_id starts.

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
16	0	2020-05-31

When customers upgrade their account from a basic plan to a pro or annual pro plan - the higher plan will take effect straightaway.

When customers churn

- they will keep their access until the end of their current billing period but the start_date will be technically the day they decided to cancel their service.

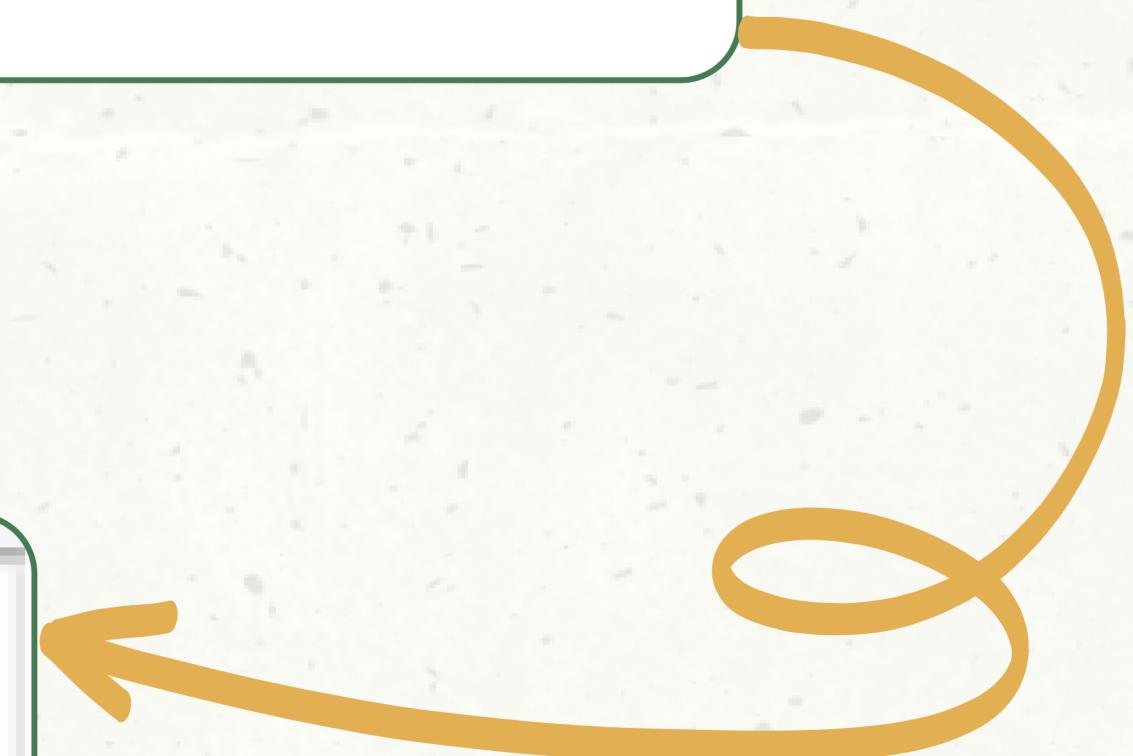
Data Analysis Questions

- How many customers has Foodie-Fi ever had?
- 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
- What plan start_date values occur after the year 2020 for our dataset? Show the breakdown by count of events for each plan_name
- What is the customer count and percentage of customers who have churned rounded to 1 decimal place?
- How many customers have churned straight after their initial free trial - what percentage is this rounded to the nearest whole number?
- What is the number and percentage of customer plans after their initial free trial?
- What is the customer count and percentage breakdown of all 5 plan_name values at 2020-12-31?
- How many customers have upgraded to an annual plan in 2020?
- How many days on average does it take for a customer to an annual plan from the day they join Foodie-Fi?
- Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)
- How many customers downgraded from a pro monthly to a basic monthly plan in 2020?

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

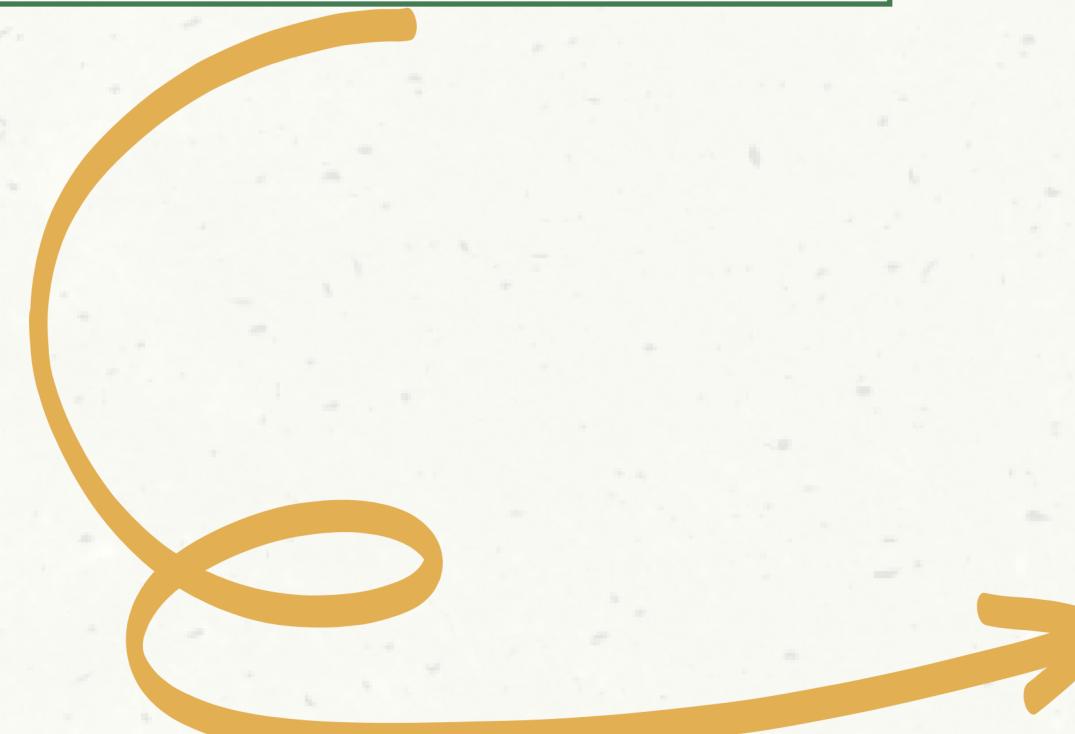
```
select count(distinct customer_id) as total_customers from subscriptions;
```

	total_customers
▶	1000



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  
    month(start_date) as month ,  
    monthname(start_date) as month_name ,  
    count(customer_id) as trial_plan_distribution  
from subscriptions  
where plan_id= 0  
group by month(start_date),monthname(start_date)  
order by month(start_date);
```



month	month_name	trial_plan_distribution
1	January	88
2	February	68
3	March	94
4	April	81
5	May	88
6	June	79
7	July	89
8	August	88
9	September	87
10	October	79
11	November	75
12	December	84

```

create view joined_tables as
(
  select
    s.customer_id,
    s.plan_id AS subscription_plan_id, -- alias to avoid duplicate column name
    s.start_date,
    p.*
  from subscriptions s
    join
    plans p on s.plan_id = p.plan_id
);
select * from joined_tables;

```

customer_id	subscription_plan_id	start_date	plan_id	plan_name	price
1	0	2020-08-01	0	trial	0.00
1	1	2020-08-08	1	basic monthly	9.90
2	0	2020-09-20	0	trial	0.00
2	3	2020-09-27	3	pro annual	199.00
3	0	2020-01-13	0	trial	0.00
3	1	2020-01-20	1	basic monthly	9.90
4	0	2020-01-17	0	trial	0.00
4	1	2020-01-24	1	basic monthly	9.90
4	4	2020-04-21	4	churn	NULL
5	0	2020-08-03	0	trial	0.00
5	1	2020-08-10	1	basic monthly	9.90
6	0	2020-12-23	0	trial	0.00

- i have created a view named **joined_tables** since there is a need in numerous queries to join both the tables

3. Show the breakdown by count of events for each plan_name after the year 2020 for our dataset?

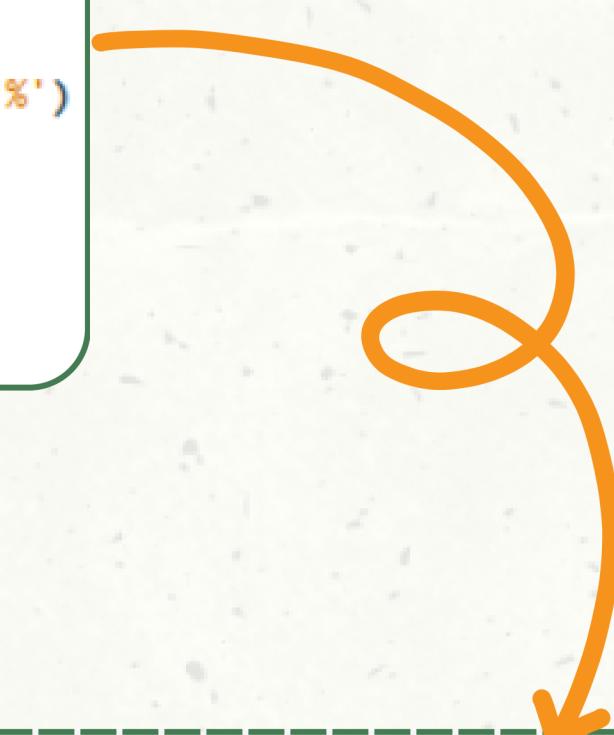
```
select  
    p.plan_name ,  
    count(s.customer_id) as total_customers  
from subscriptions s  
    join  
    plans p on s.plan_id = p.plan_id  
where year(start_date) > 2020  
group by p.plan_name ,p.plan_id  
order by p.plan_id;
```



plan_name	total_customers
basic monthly	8
pro monthly	60
pro annual	63
churn	71

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

```
select  
    count(customer_id) ,  
    concat(round(count(customer_id)/ (select count( distinct customer_id) from subscriptions) * 100, 1) ,'%')  
        as churn_percentage  
from joined_tables  
where plan_id=4 ;
```



	count(customer_id)	churn_percentage
▶	307	30.7%

5. 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 * from
    (select *,  

     row_number() over (partition by customer_id) as rn  

      from joined_tables  

    ) x
)
select
  count(customer_id) as churned_customers_after_trial ,
  concat(round(count(customer_id)/
    (select count( distinct customer_id) from subscriptions) * 100, 1) ,'%') as percentage
from cte
where customer_id in (select customer_id from cte where rn=1 and plan_name = 'trial')and
rn = 2 and plan_name = 'churn';
```



	churned_customers_after_trial	percentage
▶	92	9.2%

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

```
with cte2 as (
    select * from
        (select *,
            row_number() over (partition by customer_id) as rn
        from joined_tables) x
)
select
    count(distinct customer_id) as count_distribution,
    plan_name,
    concat(round(count(customer_id)/
        (select count( distinct customer_id) from subscriptions) * 100, 1) ,'%') as percentage
from cte2
where rn =2
group by plan_name;
```



	count_distribution	plan_name	percentage
▶	546	basic monthly	54.6%
	92	churn	9.2%
	37	pro annual	3.7%
	325	pro monthly	32.5%

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

```
with cte3 as (
    select * from
        (select *,
            row_number() over (partition by customer_id order by start_date desc) as rn
        from joined_tables
        where start_date <= "2020-12-31") x
)
select
    count(distinct customer_id) as count_distribution,
    plan_name,
    concat(round(count(customer_id)/
        (select count( distinct customer_id) from subscriptions) * 100, 1), '%') as percentage
from cte3
where rn =1
group by plan_name
order by percentage ;
```



	count_distribution	plan_name	percentage
▶	19	trial	1.9%
	195	pro annual	19.5%
	224	basic monthly	22.4%
	236	churn	23.6%
	326	pro monthly	32.6%

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

```
with monthly_cust as (
    select customer_id , start_date from subscriptions s
    where year(start_date) = "2020" and plan_id in (1,2)
),
annual_cust as (
    select customer_id , start_date from subscriptions s
    where year(start_date) = "2020" and plan_id = 3
)
select
    count(distinct a.customer_id ) as upgraded_cust
    from monthly_cust m
    join annual_cust a on a.customer_id = m.customer_id and a.start_date > m.start_date ;
```

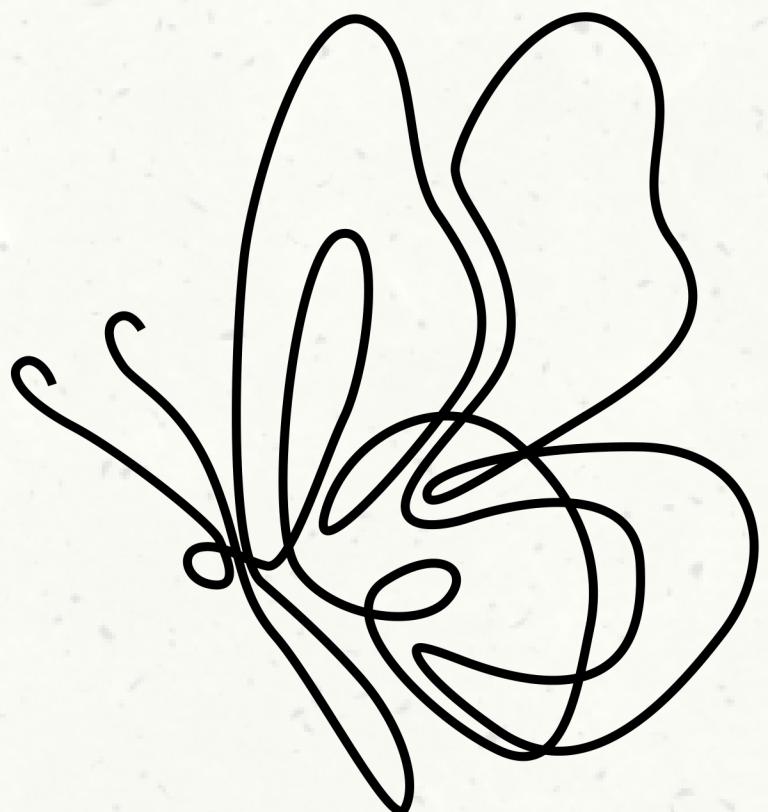


	upgraded_cust	year
▶	158	2020

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

```
with initial_plan as (
    select * from subscriptions s where plan_id=0),
annual_plan as
    (select * from subscriptions where plan_id = 3)

select avg(datediff(a.start_date , i.start_date)) as avg_date_required
from annual_plan a
join initial_plan i on a.customer_id = i.customer_id and a.start_date>i.start_date ;
```



A dashed green rounded rectangle surrounds a table. The table has two columns and two rows. The top row contains the header "avg_date_required". The bottom row contains the value "104.6202". An orange arrow points from the text "avg_date_required" in the SQL code above to the header of the table below.

	avg_date_required
▶	104.6202

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

```
with initial_plan as (
    select * from subscriptions s where plan_id=0),
annual_plan as
    (select * from subscriptions where plan_id = 3) ,
part as (select datediff(a.start_date , i.start_date) as avg_date_required,
case
    when datediff(a.start_date , i.start_date) <30 then "0-30"
    when datediff(a.start_date , i.start_date) > 30 and
        datediff(a.start_date , i.start_date) < 60 then "30-60"
    when datediff(a.start_date , i.start_date) > 60 and
        datediff(a.start_date , i.start_date) < 90 then "60-90"
    when datediff(a.start_date , i.start_date) > 90 and
        datediff(a.start_date , i.start_date) < 120 then "90-120"
    when datediff(a.start_date , i.start_date) > 120 and
        datediff(a.start_date , i.start_date) < 150 then "120-150"
    else "150 days above"
end as Part_ition
from annual_plan a
join initial_plan i on a.customer_id = i.customer_id and a.start_date>i.start_date)
select count(*) as cust_distribution , part_ition from part group by part_ition
order by part_ition;
```



	cust_distribution	Part_ition
48		0-30
42		120-150
77		150 days above
24		30-60
33		60-90
34		90-120

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

```
with pro_monthly_cust as (
    select customer_id , start_date from subscriptions s
    where year(start_date) = "2020" and plan_id = 2),
basic_monthly_cust as (select customer_id , start_date from subscriptions s
    where year(start_date) = "2020" and plan_id = 1)

select count(*) as downgraded_cust
from pro_monthly_cust a
join
basic_monthly_cust c on a.customer_id = c.customer_id
and a.start_date < c.start_date ;
```



	downgraded_cust
	0



THANK YOU!