Case Study – Foodie Fi

Krishna Murthy

Problem Statement:

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.

Datasets used

plans table: Customers can choose which plans to join Foodie-Fi when they first sign up.

There are 5 customer plans.

- Basic plan - customers have limited access and can only stream their videos and is only available monthly at $9.90
- Pro plan - customers have no watch time limits and are able to download videos for offline viewing. Pro plans start at $19.90 a month or $199 for an annual subscription.

- Trial plan - 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.
- Churn plan - When customers cancel their Foodie-Fi service - they will have a churn plan record with a null price but their plan will continue until the end of the billing period.

subscriptions table

- Customer subscriptions show the *exact date where their specific plan_id starts*.
- If customers *downgrade* from a pro plan or *cancel their subscription* - the higher plan will remain in place until the period is over - the start_date in the subscriptions table will reflect the date that the actual plan changes.
- 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

Entity Relationship Diagram:

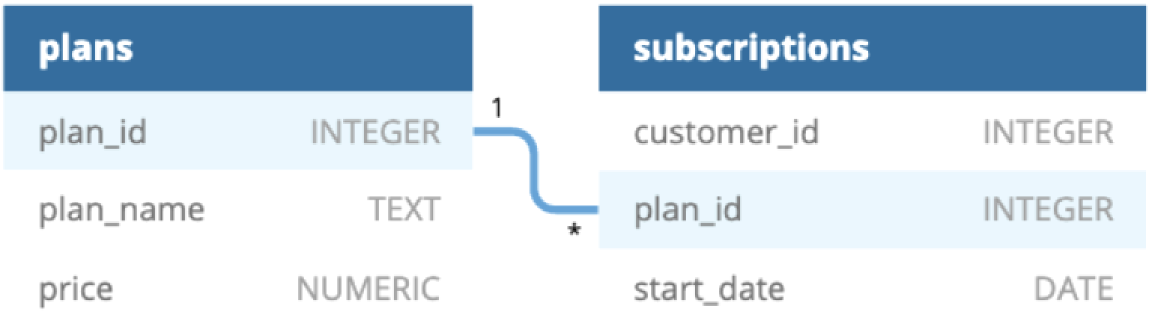
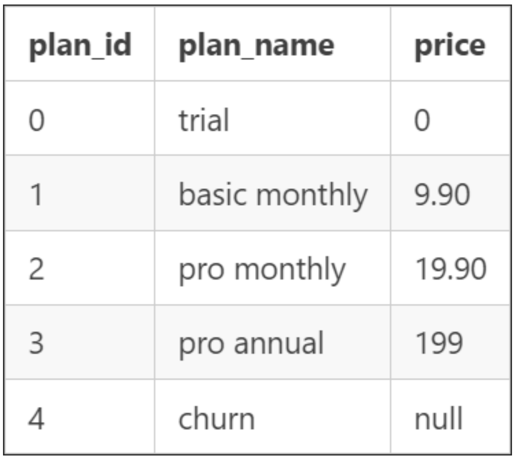
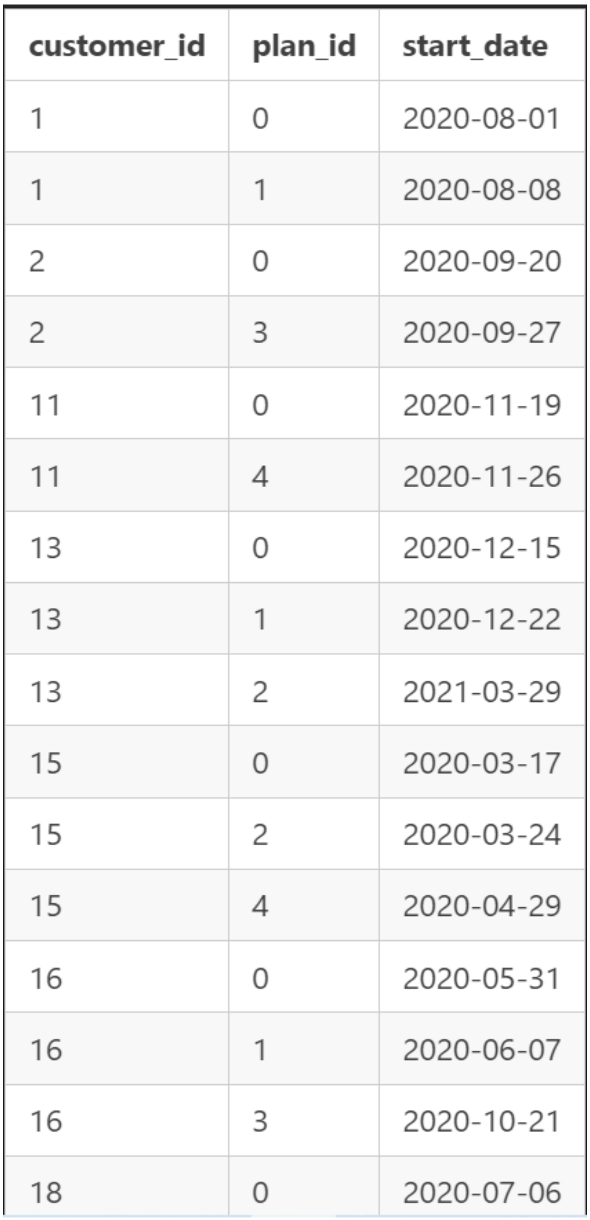


Table 1: plans



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



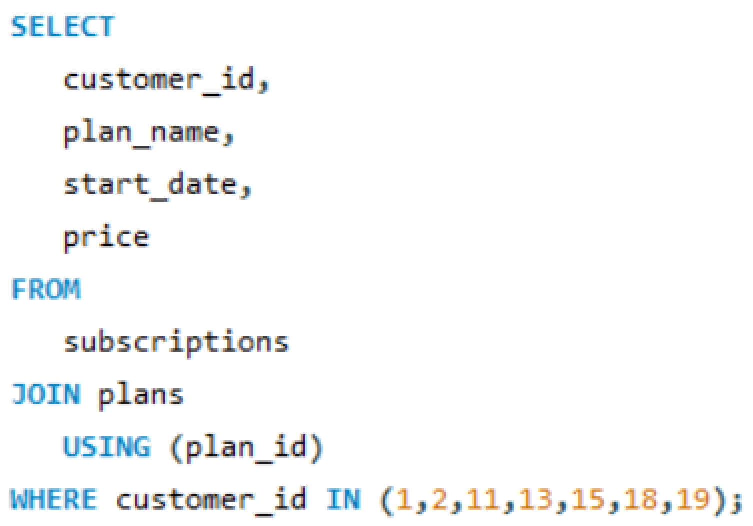
customer_id	plan_id	start_date
1	0	2020-08-01
1	1	2020-08-08
2	0	2020-09-20
2	3	2020-09-27
11	0	2020-11-19
11	4	2020-11-26
13	0	2020-12-15
13	1	2020-12-22
13	2	2021-03-29
15	0	2020-03-17
15	2	2020-03-24
15	4	2020-04-29
16	0	2020-05-31
16	1	2020-06-07
16	3	2020-10-21
18	0	2020-07-06

(Showing few records only)

CASE STUDY QUESTIONS

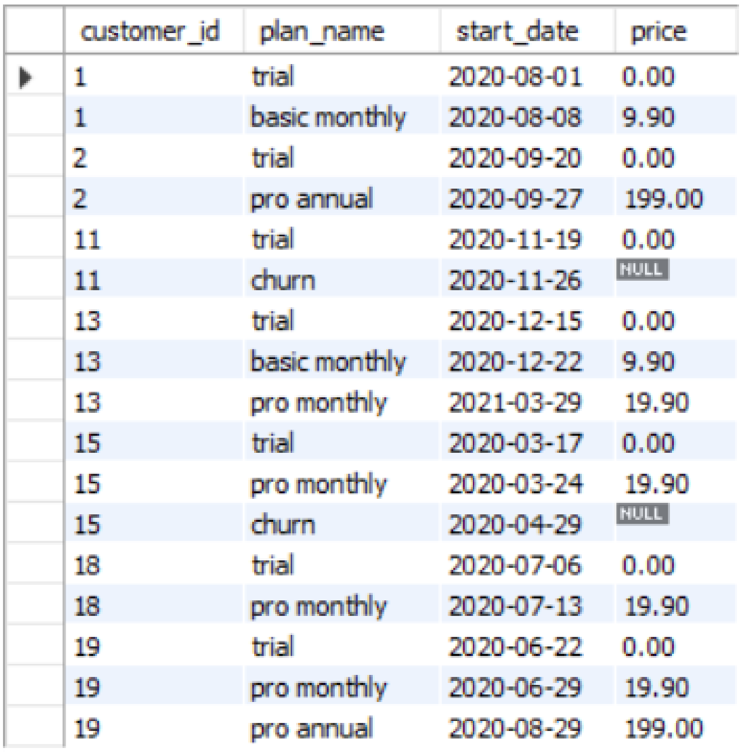
A. Customer Journey

1. **Based off the 8 sample customers provided in the sample from the subscriptions table, write a brief description about each customer's onboarding journey.**



```
SELECT
    customer_id,
    plan_name,
    start_date,
    price
FROM
    subscriptions
JOIN plans
    USING (plan_id)
WHERE customer_id IN (1,2,11,13,15,18,19);
```

Answer:



	customer_id	plan_name	start_date	price
▶	1	trial	2020-08-01	0.00
	1	basic monthly	2020-08-08	9.90
	2	trial	2020-09-20	0.00
	2	pro annual	2020-09-27	199.00
	11	trial	2020-11-19	0.00
	11	churn	2020-11-26	NULL
	13	trial	2020-12-15	0.00
	13	basic monthly	2020-12-22	9.90
	13	pro monthly	2021-03-29	19.90
	15	trial	2020-03-17	0.00
	15	pro monthly	2020-03-24	19.90
	15	churn	2020-04-29	NULL
	18	trial	2020-07-06	0.00
	18	pro monthly	2020-07-13	19.90
	19	trial	2020-06-22	0.00
	19	pro monthly	2020-06-29	19.90
	19	pro annual	2020-08-29	199.00

Customer 1:

- Customer 1 started the free trial on 1st August, 2020.
- They subscribed to the basic monthly plan after the seven day the trial period to carry on with their subscription.

Customer 2:

- Customer 2 started the free trial on 20th September, 2020.
- They subscribed to the Pro annual plan after the seven day the trial period to carry on with their subscription.

Customer 11:

- Customer 11 started the free trial on 19th November, 2020.
- They chose not to continue with paid subscription and decided to cancel on the last day of the trial period.

Customer 13:

- Customer 13 started the free trial on 15th December, 2020.
- They may have chosen to carry on with the basic monthly after the seven day the trial period.
- They then upgraded to the pro monthly plan in 29th March, 2021.

Customer 15:

- Customer 15 has chosen to start with the Pro Monthly plan on 24th March, 2020 after the free trial.
- They chose not to continue with paid subscription and decided to cancel on the last day of the Pro Monthly plan.

Customer 18:

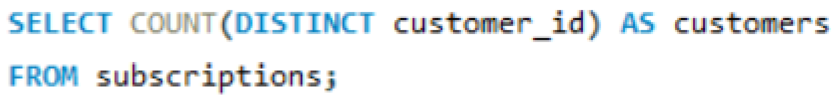
- Customer 18 started the free trial on 06th July, 2020.
- They have chosen to carry on with the Pro Monthly after the seven day the trial period.

Customer 19:

- Customer 19 started the free trial on 22nd June, 2020.
- They may have chosen to continue with the pro monthly after the seven day the trial period.
- They then upgraded to the pro annual plan in the end of August 2020.

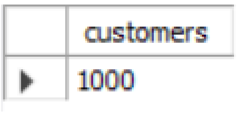
B. Data Analysis Questions

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



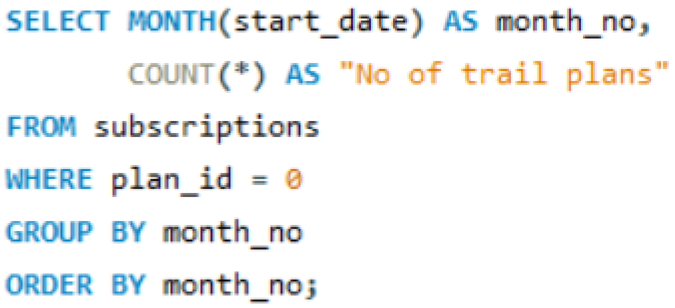
```
SELECT COUNT(DISTINCT customer_id) AS customers
FROM subscriptions;
```

Answer:



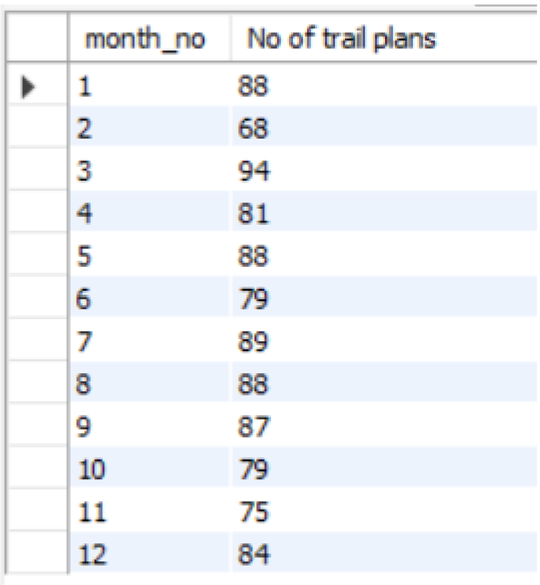
customers
1000

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



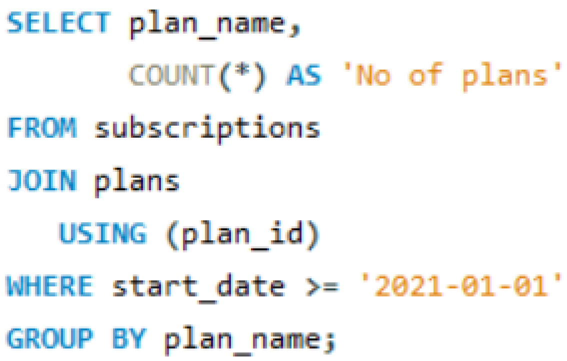
```
SELECT MONTH(start_date) AS month_no,
       COUNT(*) AS "No of trail plans"
FROM subscriptions
WHERE plan_id = 0
GROUP BY month_no
ORDER BY month_no;
```

Answer:



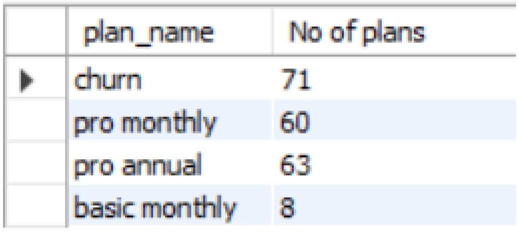
month_no	No of trail plans
1	88
2	68
3	94
4	81
5	88
6	79
7	89
8	88
9	87
10	79
11	75
12	84

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.



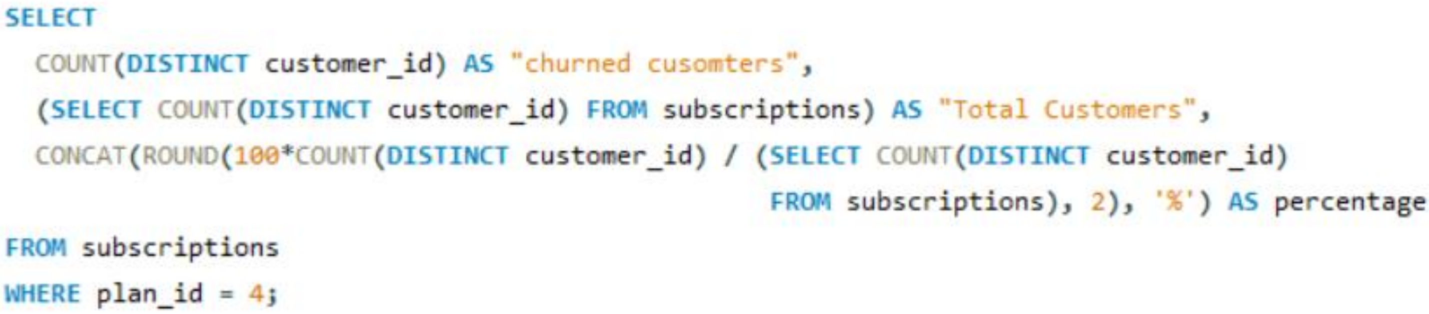
```sql
SELECT plan_name,
       COUNT(*) AS 'No of plans'
FROM subscriptions
JOIN plans
   USING (plan_id)
WHERE start_date >= '2021-01-01'
GROUP BY plan_name;
```

Answer:



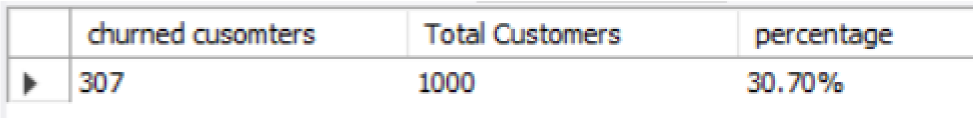
	plan_name	No of plans
▶	churn	71
	pro monthly	60
	pro annual	63
	basic monthly	8

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



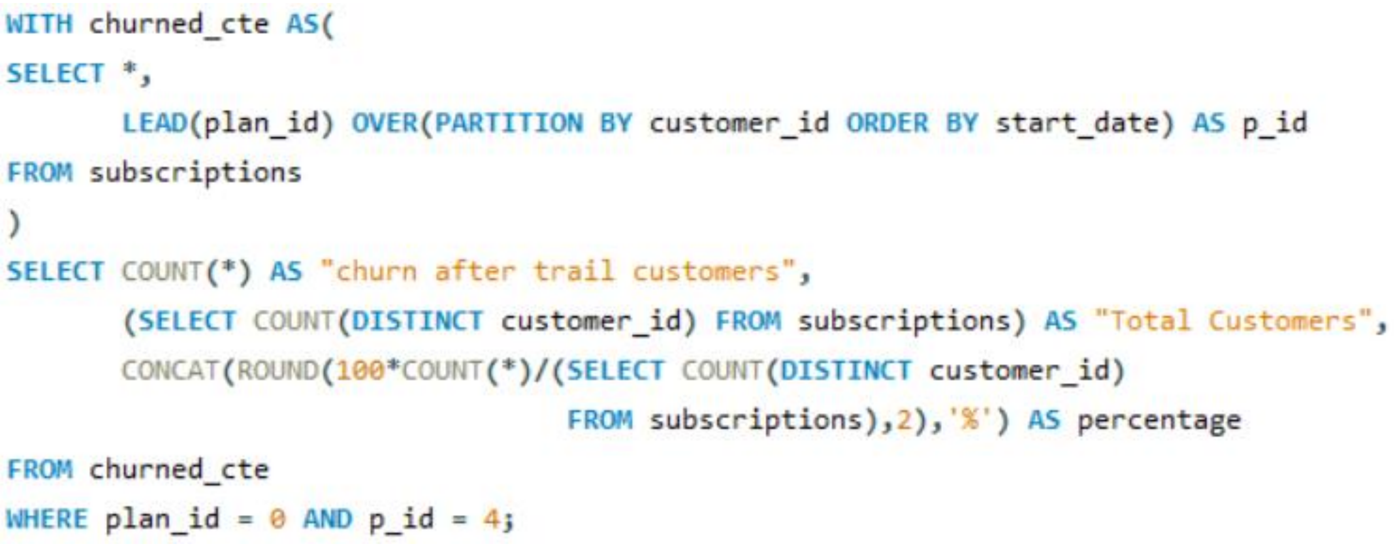
```sql
SELECT
  COUNT(DISTINCT customer_id) AS "churned cusomters",
  (SELECT COUNT(DISTINCT customer_id) FROM subscriptions) AS "Total Customers",
  CONCAT(ROUND(100*COUNT(DISTINCT customer_id) / (SELECT COUNT(DISTINCT customer_id)
                                                   FROM subscriptions), 2), '%') AS percentage
FROM subscriptions
WHERE plan_id = 4;
```

Answer:



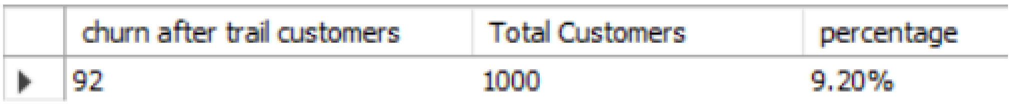
	churned cusomters	Total Customers	percentage
▶	307	1000	30.70%

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



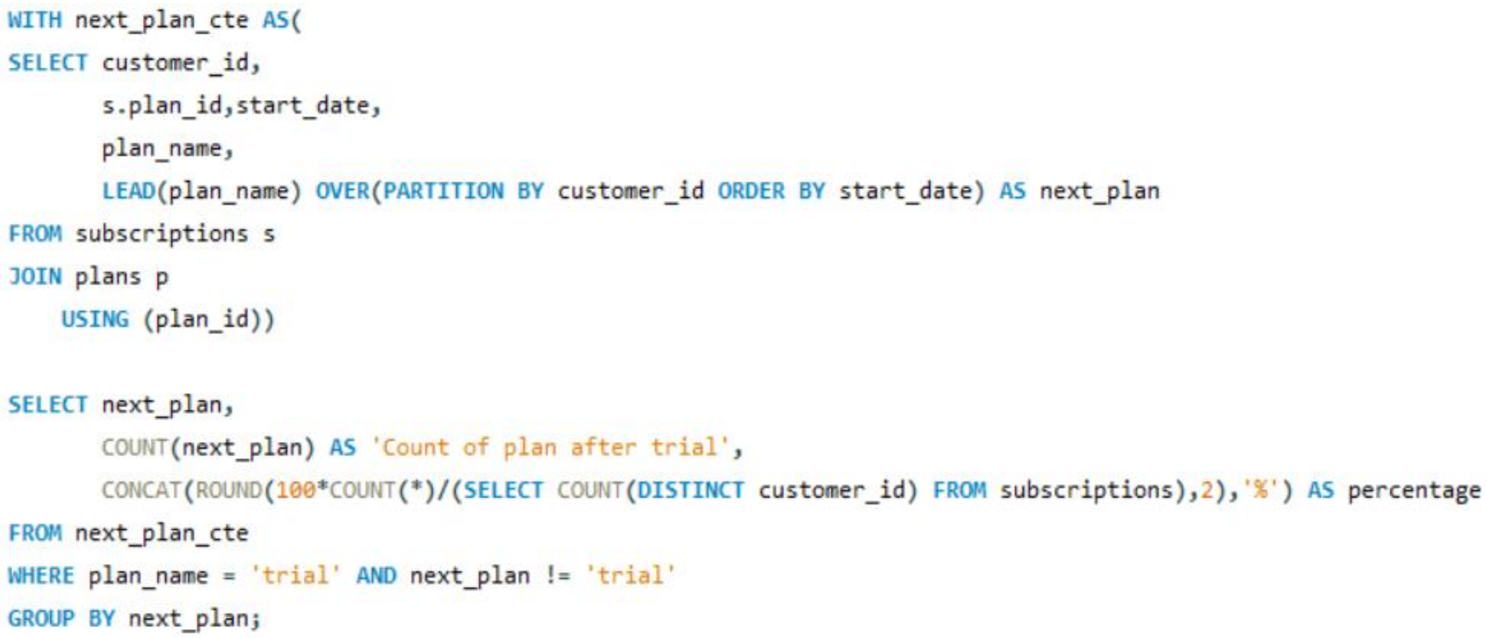
```
WITH churned_cte AS(
SELECT *,
       LEAD(plan_id) OVER(PARTITION BY customer_id ORDER BY start_date) AS p_id
FROM subscriptions
)
SELECT COUNT(*) AS "churn after trail customers",
       (SELECT COUNT(DISTINCT customer_id) FROM subscriptions) AS "Total Customers",
       CONCAT(ROUND(100*COUNT(*)/(SELECT COUNT(DISTINCT customer_id)
                                 FROM subscriptions),2),'%') AS percentage
FROM churned_cte
WHERE plan_id = 0 AND p_id = 4;
```

Answer:



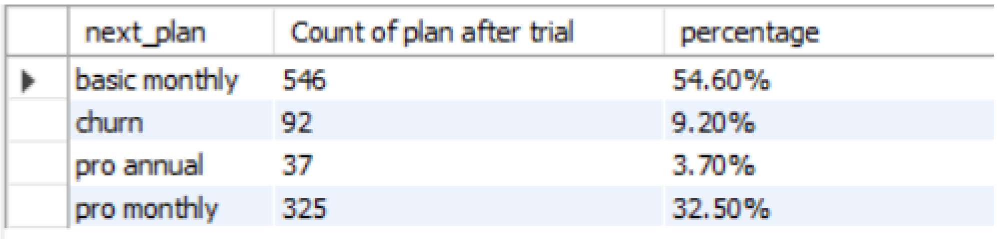
	churn after trail customers	Total Customers	percentage
▶	92	1000	9.20%

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



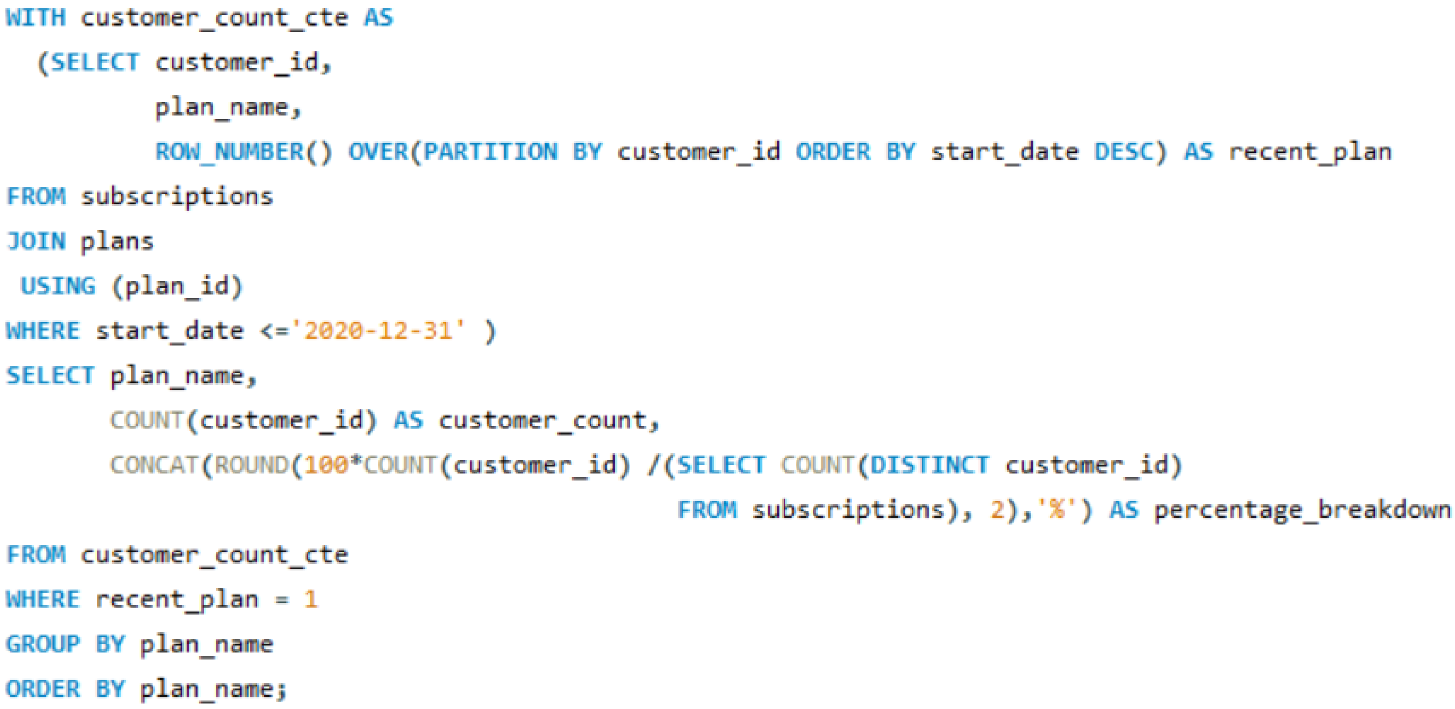
```
WITH next_plan_cte AS(
SELECT customer_id,
       s.plan_id,start_date,
       plan_name,
       LEAD(plan_name) OVER(PARTITION BY customer_id ORDER BY start_date) AS next_plan
FROM subscriptions s
JOIN plans p
    USING (plan_id))

SELECT next_plan,
       COUNT(next_plan) AS 'Count of plan after trial',
       CONCAT(ROUND(100*COUNT(*)/(SELECT COUNT(DISTINCT customer_id) FROM subscriptions),2),'%') AS percentage
FROM next_plan_cte
WHERE plan_name = 'trial' AND next_plan != 'trial'
GROUP BY next_plan;
```

Answer:



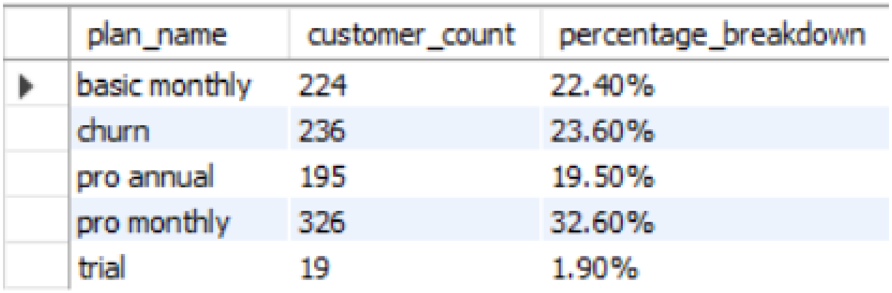
	next_plan	Count of plan after trial	percentage
▶	basic monthly	546	54.60%
	churn	92	9.20%
	pro annual	37	3.70%
	pro monthly	325	32.50%

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



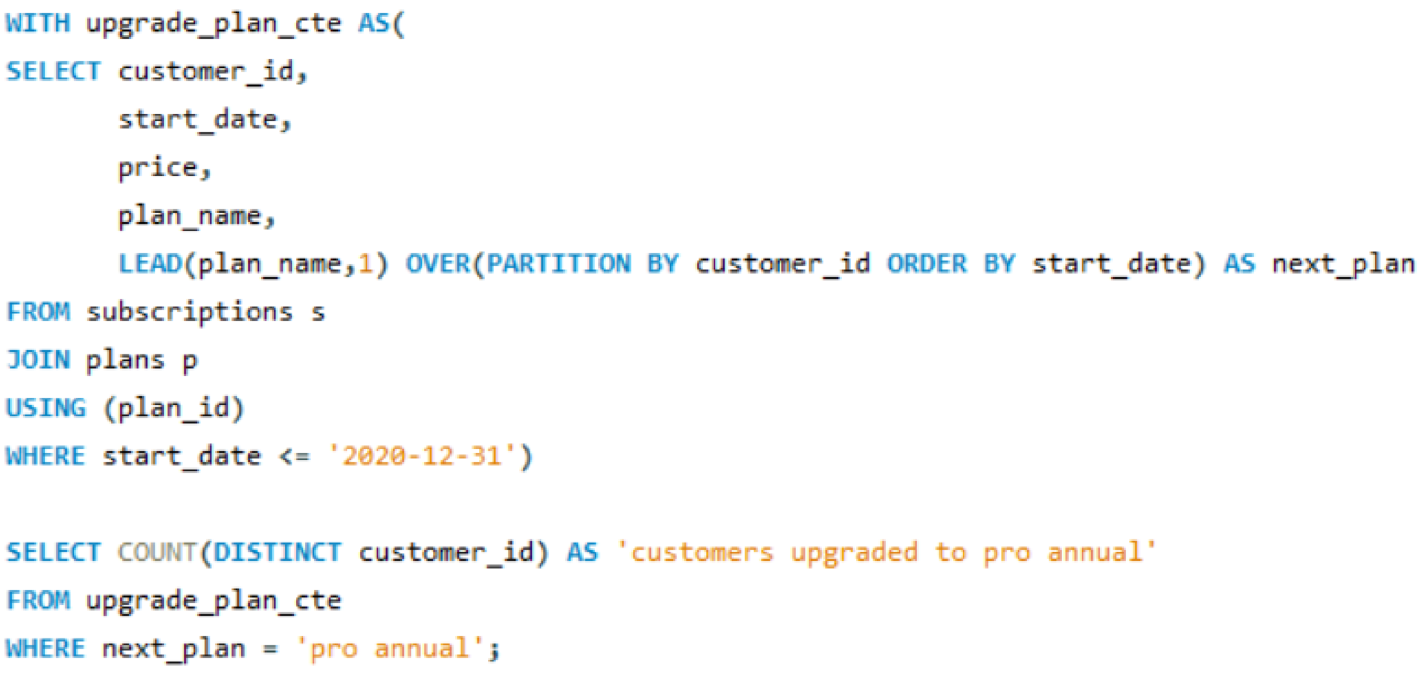
```
WITH customer_count_cte AS
  (SELECT customer_id,
          plan_name,
          ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY start_date DESC) AS recent_plan
FROM subscriptions
JOIN plans
 USING (plan_id)
WHERE start_date <='2020-12-31' )
SELECT plan_name,
       COUNT(customer_id) AS customer_count,
       CONCAT(ROUND(100*COUNT(customer_id) /(SELECT COUNT(DISTINCT customer_id)
                                            FROM subscriptions), 2),'%') AS percentage_breakdown
FROM customer_count_cte
WHERE recent_plan = 1
GROUP BY plan_name
ORDER BY plan_name;
```

Answer:



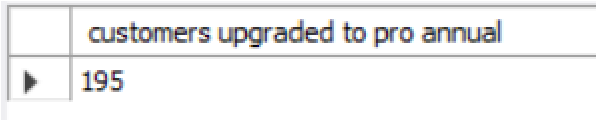
plan_name	customer_count	percentage_breakdown
basic monthly	224	22.40%
churn	236	23.60%
pro annual	195	19.50%
pro monthly	326	32.60%
trial	19	1.90%

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



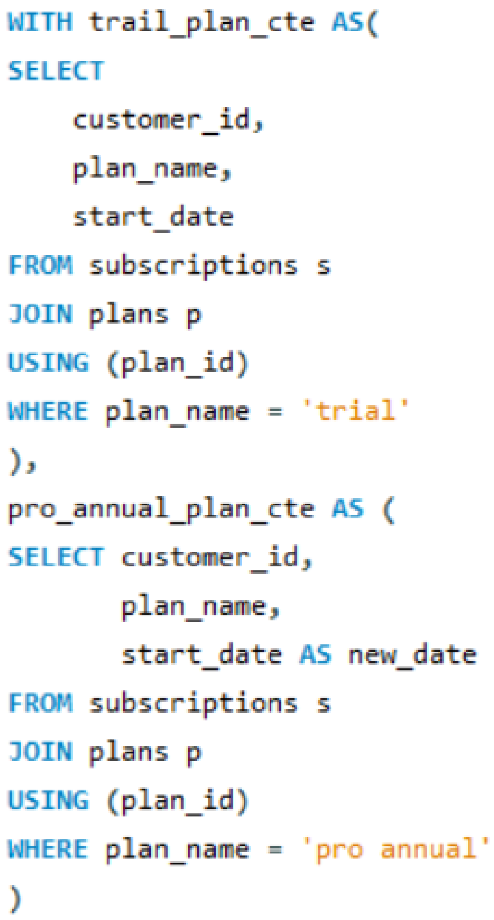
```sql
WITH upgrade_plan_cte AS(
SELECT customer_id,
       start_date,
       price,
       plan_name,
       LEAD(plan_name,1) OVER(PARTITION BY customer_id ORDER BY start_date) AS next_plan
FROM subscriptions s
JOIN plans p
USING (plan_id)
WHERE start_date <= '2020-12-31')

SELECT COUNT(DISTINCT customer_id) AS 'customers upgraded to pro annual'
FROM upgrade_plan_cte
WHERE next_plan = 'pro annual';
```

Answer:

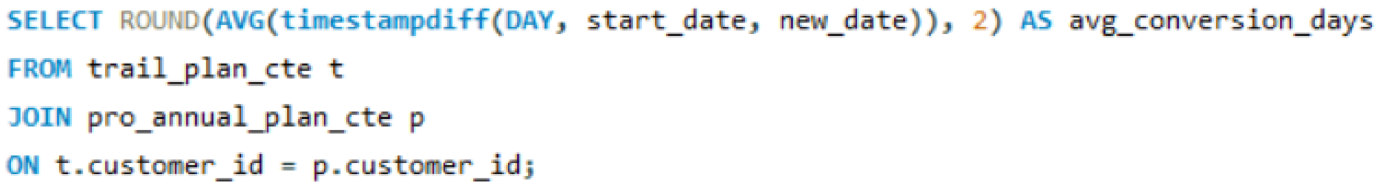


	customers upgraded to pro annual
▶	195

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

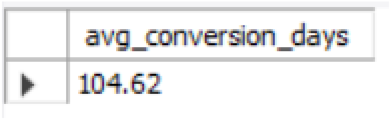


```
WITH trail_plan_cte AS(
SELECT
    customer_id,
    plan_name,
    start_date
FROM subscriptions s
JOIN plans p
USING (plan_id)
WHERE plan_name = 'trial'
),
pro_annual_plan_cte AS (
SELECT customer_id,
       plan_name,
       start_date AS new_date
FROM subscriptions s
JOIN plans p
USING (plan_id)
WHERE plan_name = 'pro annual'
)
```



```
SELECT ROUND(AVG(timestampdiff(DAY, start_date, new_date)), 2) AS avg_conversion_days
FROM trail_plan_cte t
JOIN pro_annual_plan_cte p
ON t.customer_id = p.customer_id;
```

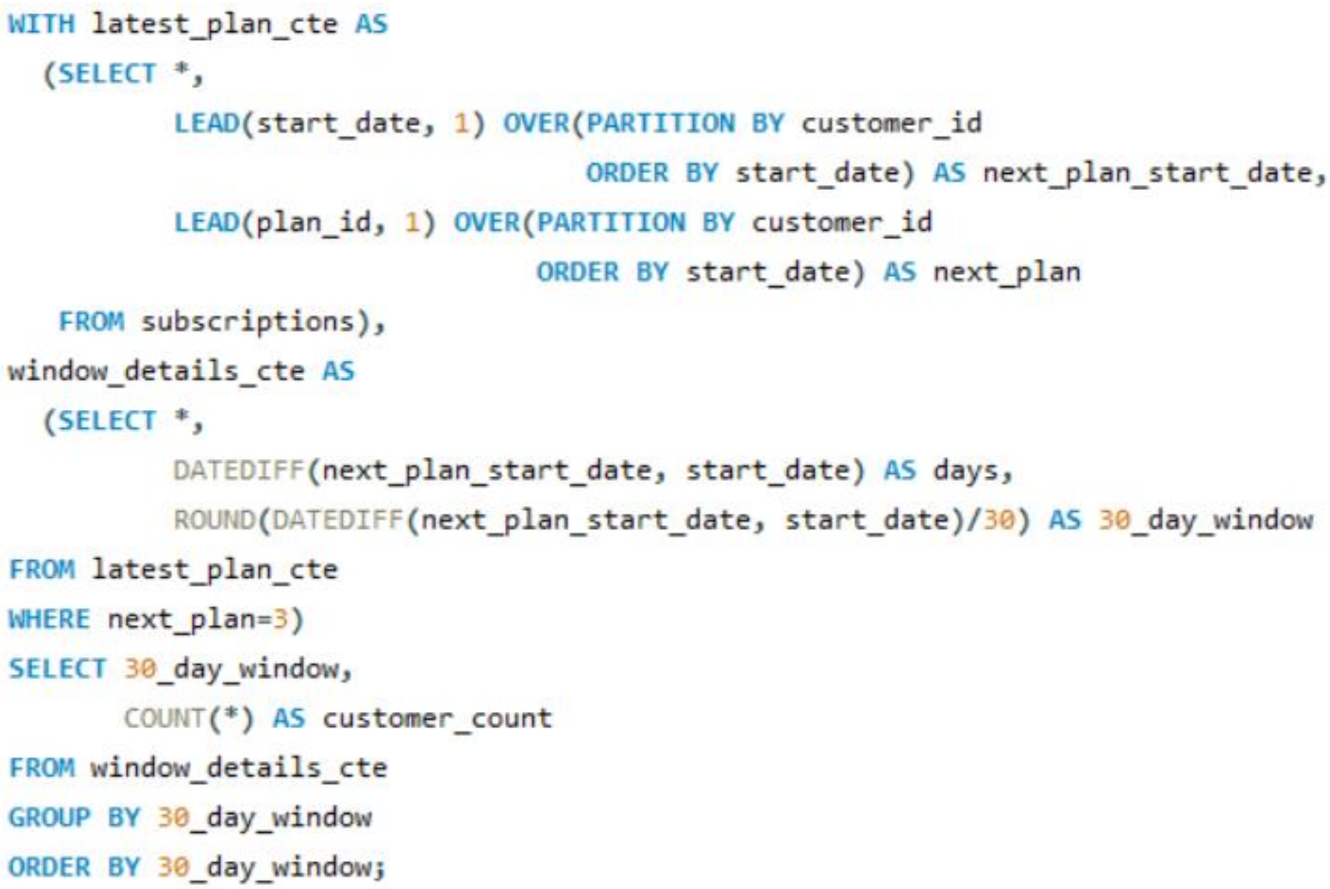
Answer:



	avg_conversion_days
▶	104.62

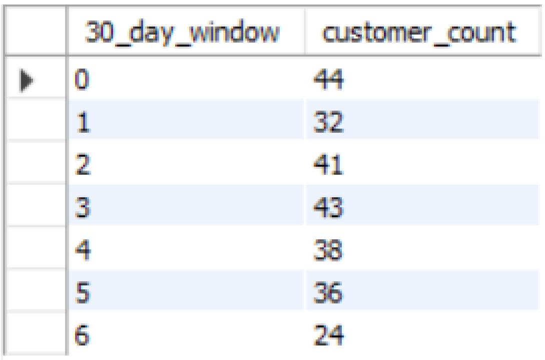
10. Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)

- The days between trial start date and the annual plan start date is computed.
- The days are bucketed in 30 day period by dividing the number of days obtained by 30.



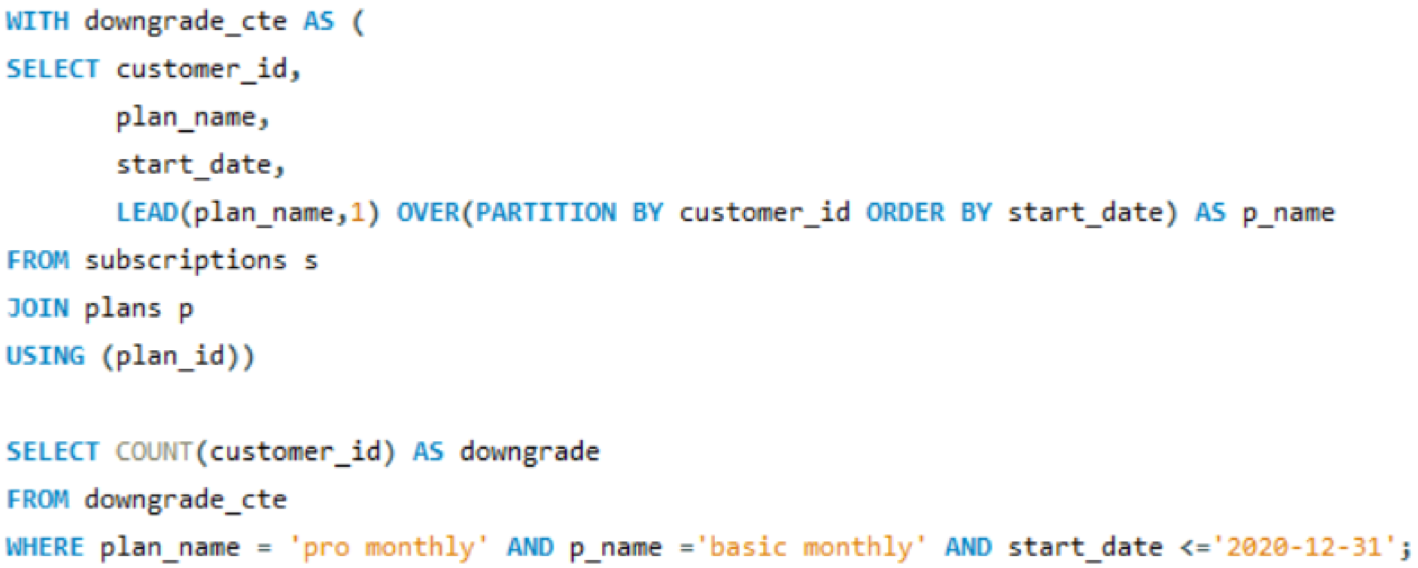
```
WITH latest_plan_cte AS
  (SELECT *,
          LEAD(start_date, 1) OVER(PARTITION BY customer_id
                                   ORDER BY start_date) AS next_plan_start_date,
          LEAD(plan_id, 1) OVER(PARTITION BY customer_id
                                 ORDER BY start_date) AS next_plan
   FROM subscriptions),
window_details_cte AS
  (SELECT *,
          DATEDIFF(next_plan_start_date, start_date) AS days,
          ROUND(DATEDIFF(next_plan_start_date, start_date)/30) AS 30_day_window
FROM latest_plan_cte
WHERE next_plan=3)
SELECT 30_day_window,
       COUNT(*) AS customer_count
FROM window_details_cte
GROUP BY 30_day_window
ORDER BY 30_day_window;
```

Answer:



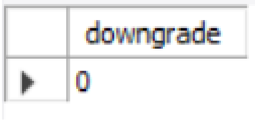
30_day_window	customer_count
0	44
1	32
2	41
3	43
4	38
5	36
6	24

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



```
WITH downgrade_cte AS (
SELECT customer_id,
       plan_name,
       start_date,
       LEAD(plan_name,1) OVER(PARTITION BY customer_id ORDER BY start_date) AS p_name
FROM subscriptions s
JOIN plans p
USING (plan_id))

SELECT COUNT(customer_id) AS downgrade
FROM downgrade_cte
WHERE plan_name = 'pro monthly' AND p_name ='basic monthly' AND start_date <='2020-12-31';
```

Answer:



	downgrade
▶	0

THANK YOU!