FOODIE – FI: A SQL CASE STUDY

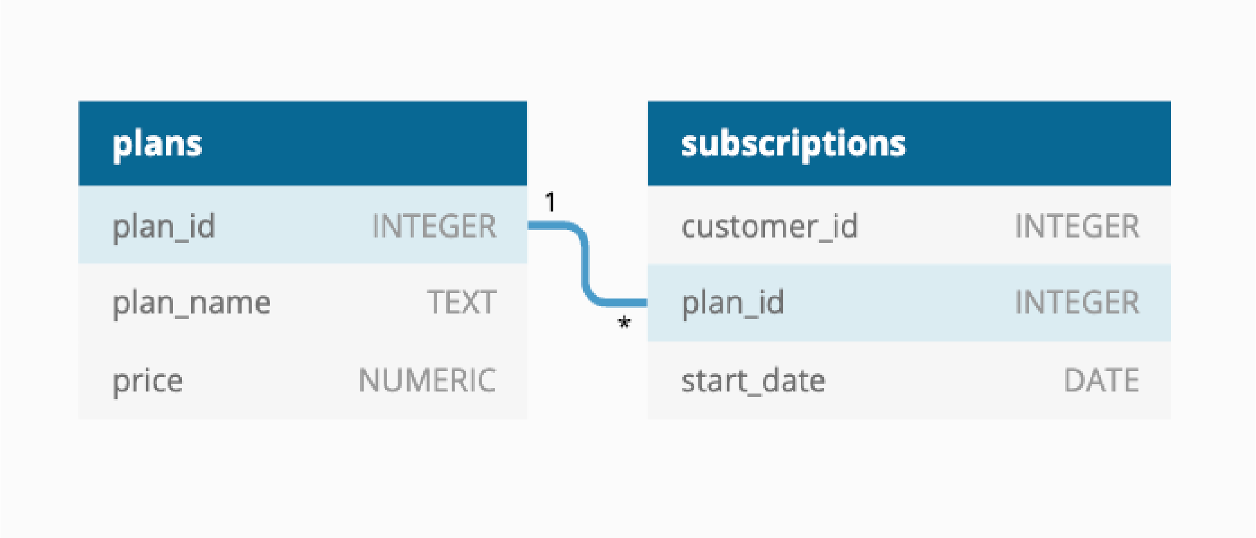
INTRODUCTION

Danny realized that 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 start-up 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!

This case study focuses on using subscription-style digital data to answer important business questions on **Customer Journeys, Payments, and Business Performances**.

Full Description

Entity Relationship Diagram



plans	
plan_id	INTEGER
plan_name	TEXT
price	NUMERIC

subscriptions	
customer_id	INTEGER
plan_id	INTEGER
start_date	DATE

Table 1: Plans

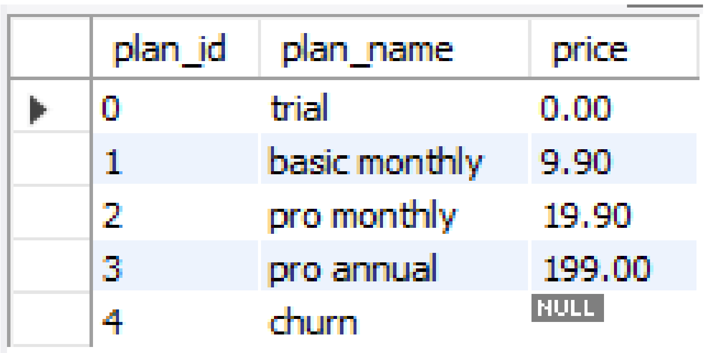
- There are 5 customer plans.
- **Trial**— Customer sign up to an initial 7-day free trial and will automatically continue with the **pro monthly** subscription plan unless they cancel, downgrade to basic, or upgrade to an annual pro plan at any point during the trial.
- **Basic plan** — Customers have 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.

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.

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

```
# Creating Schema foodie_fi
CREATE SCHEMA foodie_fi;
USE foodie_fi;
# creating table plans
DROP TABLE plans;
CREATE TABLE plans (
    plan_id INTEGER,
    plan_name TEXT,
    price DECIMAL(10,2)
);
# Inserting Values in table plans
INSERT INTO plans ( plan_id, plan_name, price)
VALUES      (0, 'trial', 0),
            (1, 'basic monthly', 9.90),
            (2, 'pro monthly', 19.90),
            (3, 'pro annual', 199),
            (4, 'churn', null);
```



plan_id	plan_name	price
0	trial	0.00
1	basic monthly	9.90
2	pro monthly	19.90
3	pro annual	199.00
4	churn	NULL

Table 2: Subscriptions

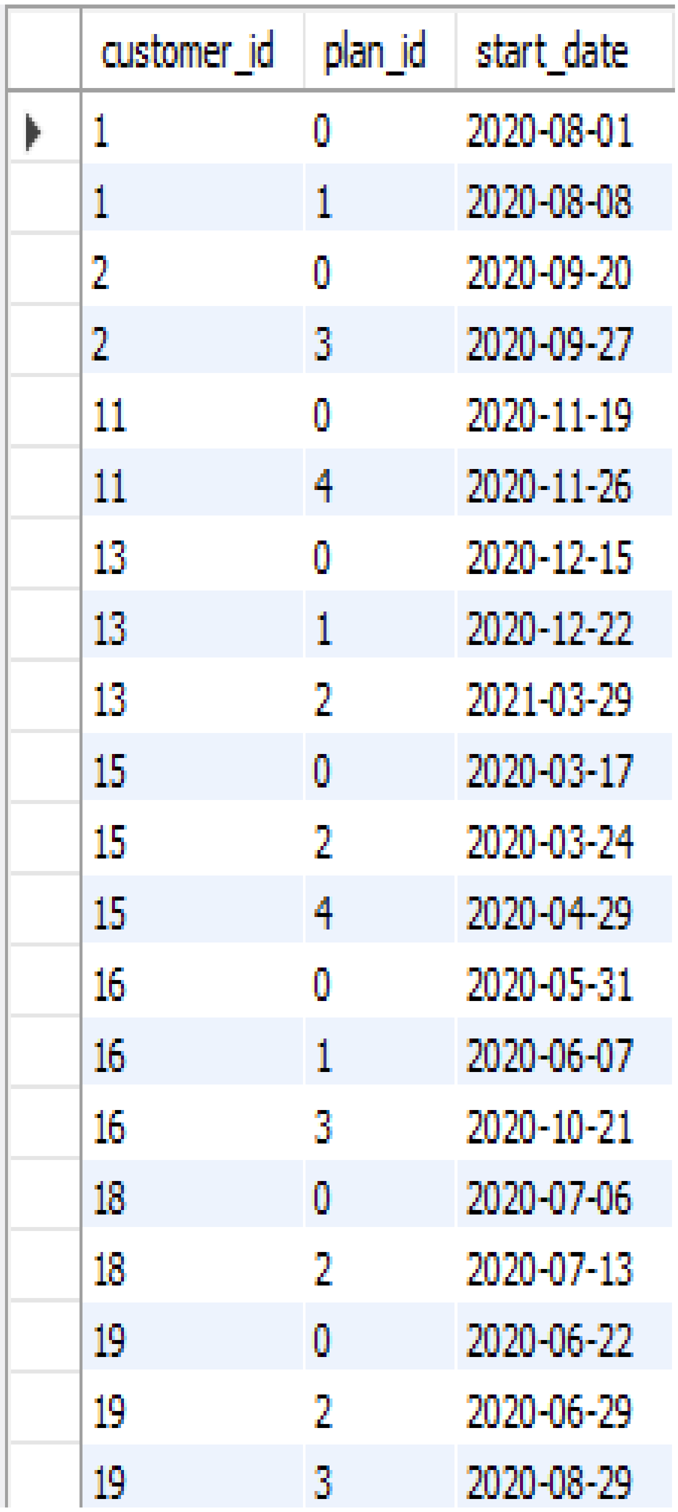
- Customer subscriptions show the **exact date** when their specific plan_id starts.
- If customers downgrade from a pro plan or cancel their subscription — the higher plan will remain in place until the period is over — the start_date in the subscriptions table will reflect the date that the actual plan changes.
- When customers upgrade their account from a basic plan to a pro or annual pro plan — the higher plan will take effect straight away.
- 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.

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

customer_id	plan_id	start_date
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
18	2	2020-07-13
19	0	2020-06-22
19	2	2020-06-29
19	3	2020-08-29

```
# Creating table subscriptions

CREATE TABLE subscriptions (

  customer_id INTEGER,

  plan_id INTEGER,

  start_date DATE

);
```

```
# Inserting values into subscriptions

INSERT INTO subscriptions

VALUES      (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'),
            (18, 2, '2020-07-13'),
            (19, 0, '2020-06-22'),
            (19, 2, '2020-06-29')
            (19, 3, '2020-08-29');
```



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
18	2	2020-07-13
19	0	2020-06-22
19	2	2020-06-29
19	3	2020-08-29

CASE STUDY QUESTIONS

A. Customer Journey

Based on the 8 sample customers provided in the sample from the subscriptions table, write a brief description of each customer on the boarding journey.

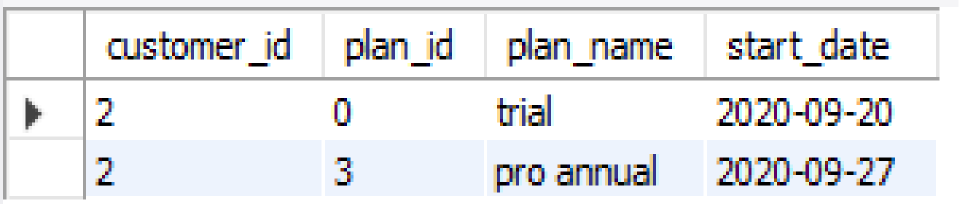
Customer 1:

```
SELECT customer_id, plan_id, plan_name, start_date

FROM subscriptions

JOIN plans USING (plan_id)

WHERE customer_id = 1;
```

Customer with ID 1 started with a trial subscription and continued with a basic monthly subscription in 7 days after sign-up.

Customer 2:

```
SELECT customer_id, plan_id, plan_name, start_date

FROM subscriptions

JOIN plans USING (plan_id)

WHERE customer_id = 2;
```

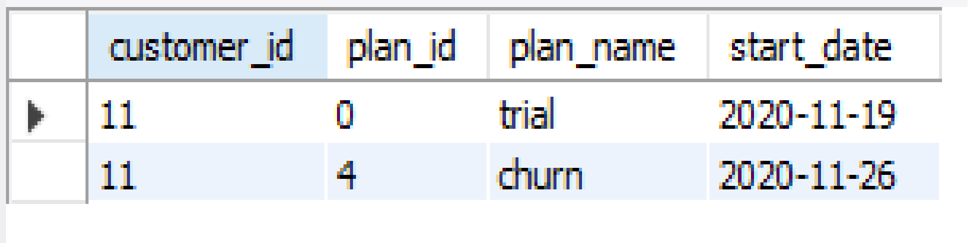


	customer_id	plan_id	plan_name	start_date
▶	2	0	trial	2020-09-20
	2	3	pro annual	2020-09-27

Customer with ID 2 started with a trial subscription and continued with a pro annual subscription 7 days after sign-up.

Customer 11:

```
SELECT customer_id, plan_id, plan_name, start_date

FROM subscriptions

JOIN plans USING (plan_id)

WHERE customer_id = 11;
```

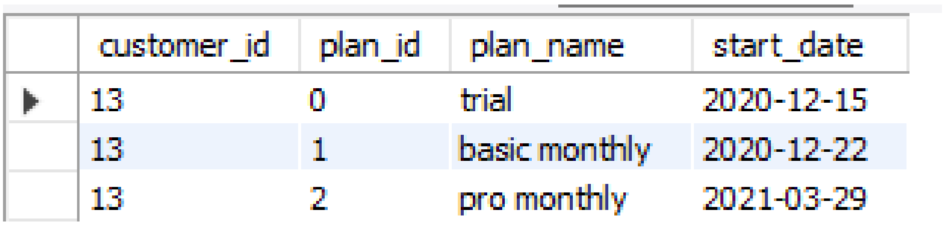


	customer_id	plan_id	plan_name	start_date
▶	11	0	trial	2020-11-19
	11	4	churn	2020-11-26

Customer with ID 11 started with a trial subscription and churned in 7 days after sign-up.

Customer 13:

```
SELECT customer_id, plan_id, plan_name, start_date

FROM subscriptions

JOIN plans USING (plan_id)

WHERE customer_id = 13;
```

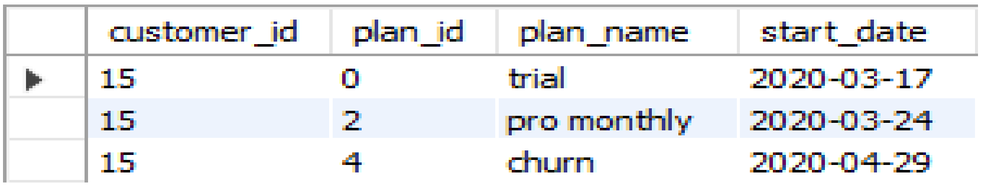


	customer_id	plan_id	plan_name	start_date
▶	13	0	trial	2020-12-15
	13	1	basic monthly	2020-12-22
	13	2	pro monthly	2021-03-29

Customer with ID 13 started with a trial subscription, then purchased a basic monthly subscription 7 days after sign-up, and after 3 months upgraded to a pro monthly subscription.

Customer 15:

```
SELECT customer_id, plan_id, plan_name, start_date

FROM subscriptions

JOIN plans USING (plan_id)

WHERE customer_id = 15;
```

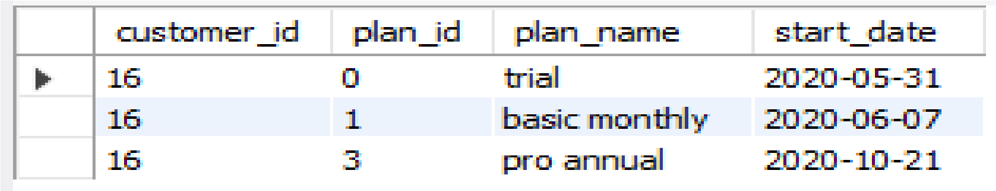


	customer_id	plan_id	plan_name	start_date
▶	15	0	trial	2020-03-17
	15	2	pro monthly	2020-03-24
	15	4	churn	2020-04-29

Customer with ID 15 started with a trial subscription, purchased a basic monthly subscription 7 days after sign-up, and churned in a month.

Customer 16:

```
SELECT customer_id, plan_id, plan_name, start_date

FROM subscriptions

JOIN plans USING (plan_id)

WHERE customer_id = 16;
```

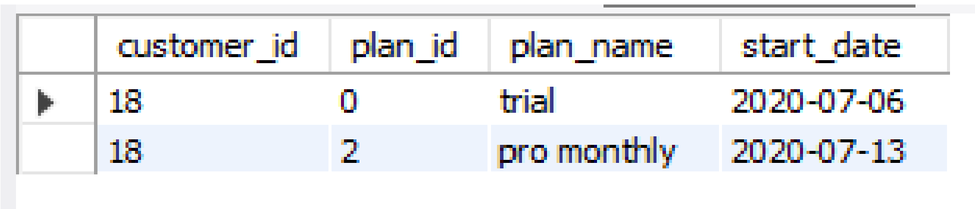


	customer_id	plan_id	plan_name	start_date
▶	16	0	trial	2020-05-31
	16	1	basic monthly	2020-06-07
	16	3	pro annual	2020-10-21

Customer with ID 16 started with a trial subscription, purchased a basic monthly subscription in 7 days after sign-up, and in 4 months after that upgraded a pro annual subscription.

Customer 18:

```
SELECT customer_id, plan_id, plan_name, start_date
FROM subscriptions
JOIN plans USING (plan_id)
WHERE customer_id = 18;
```

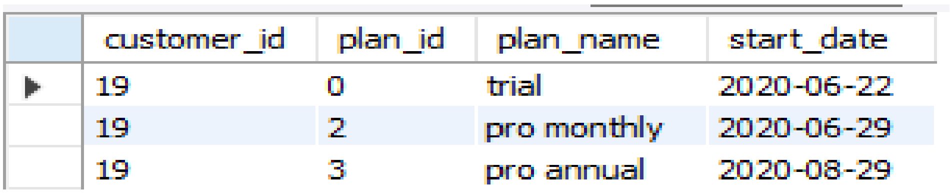


	customer_id	plan_id	plan_name	start_date
▶	18	0	trial	2020-07-06
	18	2	pro monthly	2020-07-13

Customer with ID 18 started with a trial subscription and continued with a pro monthly subscription in 7 days after sign-up.

Customer 19:

```
SELECT customer_id, plan_id, plan_name, start_date
FROM subscriptions
JOIN plans USING (plan_id)
WHERE customer_id = 19;
```



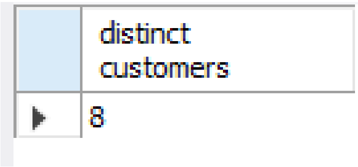
	customer_id	plan_id	plan_name	start_date
▶	19	0	trial	2020-06-22
	19	2	pro monthly	2020-06-29
	19	3	pro annual	2020-08-29

Customer with ID 19 started with a trial subscription, continued with a pro monthly subscription 7 days after sign-up, and upgraded to a pro annual subscription in 2 months.

B. DATA ANALYSIS QUESTIONS

1. **How many customers have Foodie-Fi ever had?**

```
SELECT COUNT (DISTINCT (customer_id)) AS 'distinct customers'

FROM subscriptions;
```

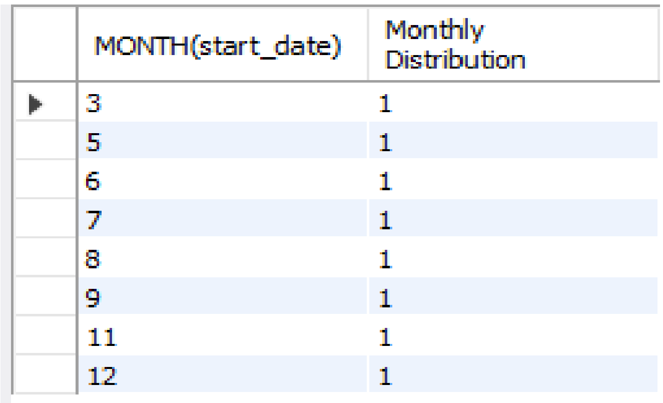


distinct customers
8

Foodie-Fi has **8** Unique Customers.

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), COUNT (DISTINCT customer_id) AS 'Monthly Distribution'

FROM subscriptions

JOIN plans USING (plan_id)

WHERE plan_id = 0

GROUP BY MONTH(start_date);
```

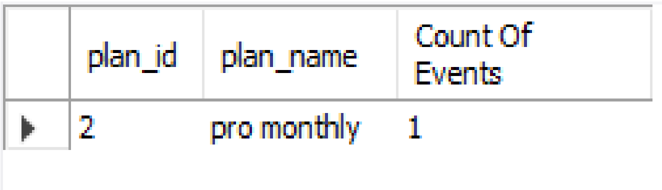


MONTH(start_date)	Monthly Distribution
3	1
5	1
6	1
7	1
8	1
9	1
11	1
12	1

The monthly Distribution of trial plans is the **same** in all months i.e. **1**, Except January, February, April, and October which has **0**.

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

```
SELECT  plan_id, plan_name, COUNT (*) AS 'Count Of Events'

FROM subscriptions

JOIN plans USING (plan_id)

WHERE YEAR(start_date) > 2020

GROUP BY plan_id;
```

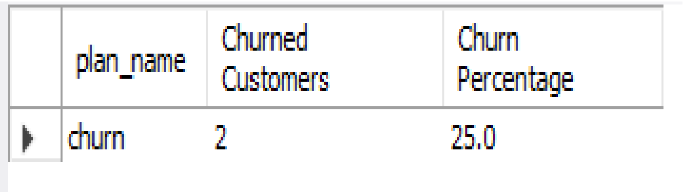


	plan_id	plan_name	Count Of Events
▶	2	pro monthly	1

The count of events after the year 2020 was found to be **1** for plan **“pro monthly”**, there were no other plans found.

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

```
SELECT

   plan_name,

   COUNT (DISTINCT customer_id) AS 'Churned Customers',

   ROUND (100 * COUNT (DISTINCT customer_id) /

   (SELECT COUNT (DISTINCT customer_id) AS 'Distinct Customers'

FROM subscriptions),1) AS 'Churn Percentage'

FROM subscriptions

JOIN plans USING (plan_id)

WHERE plan_id = 4;
```

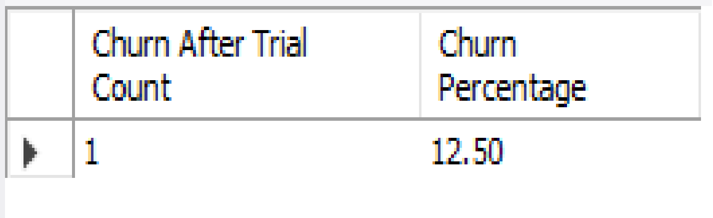


	plan_name	Churned Customers	Churn Percentage
▶	churn	2	25.0

There are **2** Customers who churned, which is **25%** of the Foodie-Fi Customer base.

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

```
WITH next_plan_cte AS
    (SELECT *,
    LEAD (plan_id, 1) OVER (PARTITION BY customer_id ORDER BY start_date) AS next_plan
FROM subscriptions), Churners AS
    (SELECT *
    FROM next_plan_cte
    WHERE next_plan = 4
   AND plan_id = 0)
SELECT COUNT(customer_id) AS 'Churn After Trial Count',
      ROUND (100 * COUNT(customer_id) / (SELECT COUNT (DISTINCT customer_id) AS 'Distinct Customers'
     FROM subscriptions),2) AS 'Churn Percentage'
FROM Churners;
```

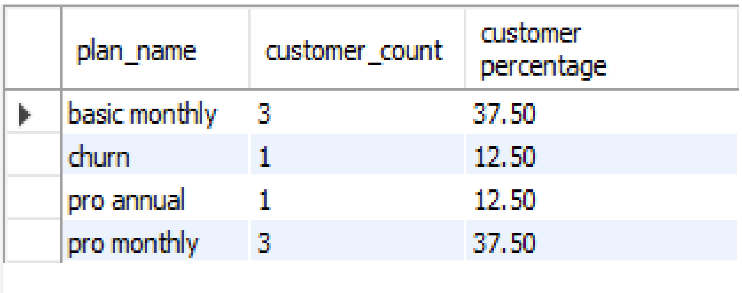


	Churn After Trial Count	Churn Percentage
▶	1	12.50

1 Customer Churned straight after the initial free trial which is **12.50%** of the entire customer base.

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

```
WITH previous_plan_cte AS
(SELECT *,
lag(plan_id, 1) over(PARTITION BY customer_id ORDER BY start_date) AS previous_plan
FROM subscriptions
JOIN plans USING (plan_id))
SELECT plan_name, count(customer_id) customer_count,
round(100 *count(DISTINCT customer_id) /

(SELECT count (DISTINCT customer_id) AS 'distinct customers'

FROM subscriptions), 2) AS 'customer percentage'

FROM previous_plan_cte

WHERE previous_plan=0

GROUP BY plan_name;
```

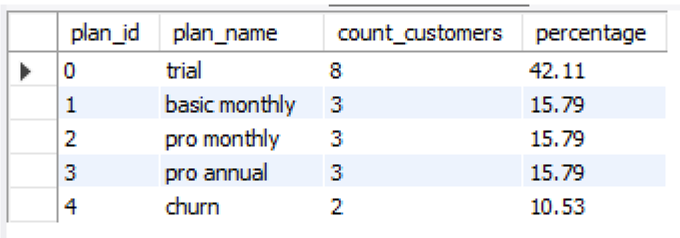


	plan_name	customer_count	customer percentage
▶	basic monthly	3	37.50
	churn	1	12.50
	pro annual	1	12.50
	pro monthly	3	37.50

More than 80% of customers are on paid plans with a medium percent of 12.50% in plan 3 (pro annual $199).

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

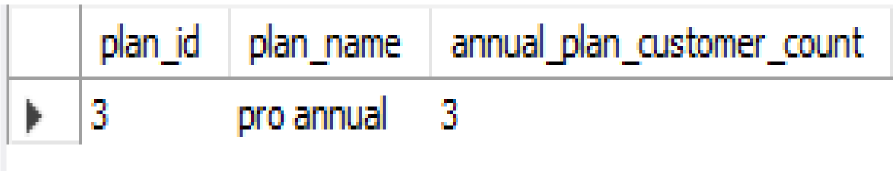
```
WITH customer_count AS (

  SELECT plan_id, plan_name, COUNT(DISTINCT customer_id) AS count_customers

  FROM subscriptions

  JOIN plans USING (plan_id)

  WHERE start_date <= '2020-12-31'

  GROUP BY plan_name

)

SELECT plan_id, plan_name, count_customers, ROUND(100*(count_customers/(SELECT SUM(count_customers) FROM customer_count)), 2) AS percentage

FROM customer_count

ORDER BY plan_id;
```



plan_id	plan_name	count_customers	percentage
0	trial	8	42.11
1	basic monthly	3	15.79
2	pro monthly	3	15.79
3	pro annual	3	15.79
4	churn	2	10.53

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

```
SELECT plan_id, plan_name,
       COUNT(DISTINCT customer_id) AS annual_plan_customer_count
FROM subscriptions
JOIN plans USING(plan_id)
WHERE plan_id = 3 AND start_date <= '2020-12-31';
```

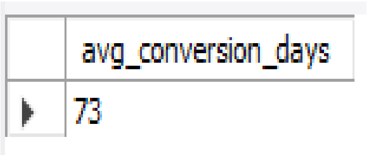


plan_id	plan_name	annual_plan_customer_count
3	pro annual	3

3 Customers upgraded to a pro 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?**

```
WITH trial_plan_customer_cte AS
  (SELECT * FROM subscriptions
     JOIN plans USING (plan_id)
     WHERE plan_id=0), annual_plan_customer_cte AS
  (SELECT * FROM subscriptions
   JOIN plans USING (plan_id)
   WHERE plan_id=3)
SELECT round(avg(datediff(annual_plan_customer_cte.start_date,
trial_plan_customer_cte.start_date)))AS avg_conversion_days
FROM trial_plan_customer_cte
INNER JOIN annual_plan_customer_cte USING (customer_id);
```



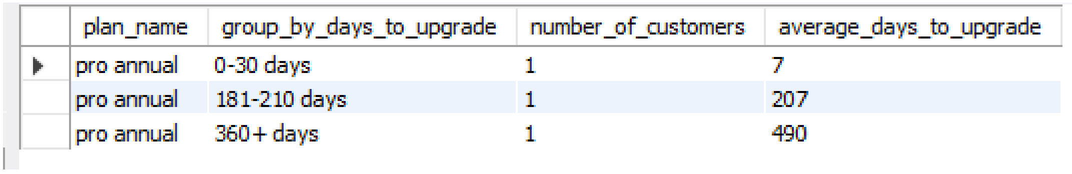
avg_conversion_days
73

On average, it takes **73** days for a customer to upgrade 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.)**

```
SELECT * FROM
 (
  SELECT
   plan_name,
   CASE
    WHEN s.start_date - t.start_date < 31 THEN '0-30 days'
    WHEN s.start_date - t.start_date BETWEEN 31
    AND 60 THEN '31-60 days'
    WHEN s.start_date - t.start_date BETWEEN 61
    AND 90 THEN '61-90 days'
    WHEN s.start_date - t.start_date BETWEEN 91
    AND 120 THEN '91-120 days'
    WHEN s.start_date - t.start_date BETWEEN 121
    AND 150 THEN '121-150 days'
    WHEN s.start_date - t.start_date BETWEEN 151
    AND 180 THEN '151-180 days'
    WHEN s.start_date - t.start_date BETWEEN 181
    AND 210 THEN '181-210 days'
    WHEN s.start_date - t.start_date BETWEEN 211
    AND 240 THEN '211-240 days'
    WHEN s.start_date - t.start_date BETWEEN 241
    AND 270 THEN '241-270 days'
    WHEN s.start_date - t.start_date BETWEEN 271
```

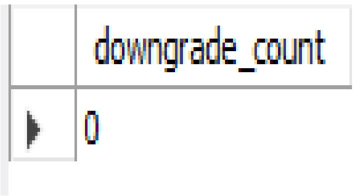
```
        AND 300 THEN '271-300 days'
        WHEN s.start_date - t.start_date BETWEEN 301
        AND 330 THEN '301-330 days'
        WHEN s.start_date - t.start_date BETWEEN 331
        AND 360 THEN '331-360 days'
        WHEN s.start_date - t.start_date > 360 THEN '360+ days'
      END AS group_by_days_to_upgrade,
      COUNT(s.start_date - t.start_date) AS number_of_customers,
      ROUND(AVG(s.start_date - t.start_date)) AS average_days_to_upgrade
    FROM
      subscriptions AS s
      JOIN plans AS p ON s.plan_id = p.plan_id
      JOIN (
        SELECT
          customer_id,
          start_date
        FROM
          subscriptions
        WHERE
          plan_id = 0
      ) AS t ON s.customer_id = t.customer_id
    WHERE
      plan_name = 'pro annual'
    GROUP BY
      plan_name,
      group_by_days_to_upgrade
  ) AS count_groups
GROUP BY
  plan_name,
  group_by_days_to_upgrade,
  number_of_customers,
  average_days_to_upgrade
ORDER BY
  CASE
    WHEN group_by_days_to_upgrade = '0-30 days' THEN 1
    WHEN group_by_days_to_upgrade = '31-60 days' THEN 2
    WHEN group_by_days_to_upgrade = '61-90 days' THEN 3
    WHEN group_by_days_to_upgrade = '91-120 days' THEN 4
    WHEN group_by_days_to_upgrade = '121-150 days' THEN 5
    WHEN group_by_days_to_upgrade = '151-180 days' THEN 6
    WHEN group_by_days_to_upgrade = '181-210 days' THEN 7
    WHEN group_by_days_to_upgrade = '211-240 days' THEN 8
    WHEN group_by_days_to_upgrade = '241-270 days' THEN 9
    WHEN group_by_days_to_upgrade = '271-300 days' THEN 10
    WHEN group_by_days_to_upgrade = '301-330 days' THEN 11
    WHEN group_by_days_to_upgrade = '331-360 days' THEN 12
    WHEN group_by_days_to_upgrade = '360+ days' THEN 13
  END;
```



plan_name	group_by_days_to_upgrade	number_of_customers	average_days_to_upgrade
pro annual	0-30 days	1	7
pro annual	181-210 days	1	207
pro annual	360+ days	1	490

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

```
WITH next_plan_cte AS
  (SELECT *,
     lead(plan_id, 1) OVER(PARTITION BY customer_id ORDER BY start_date) AS next_plan
FROM subscriptions)
SELECT count(*) AS downgrade_count
FROM next_plan_cte
WHERE plan_id=2 AND next_plan=1 AND year(start_date);
```



downgrade_count
0

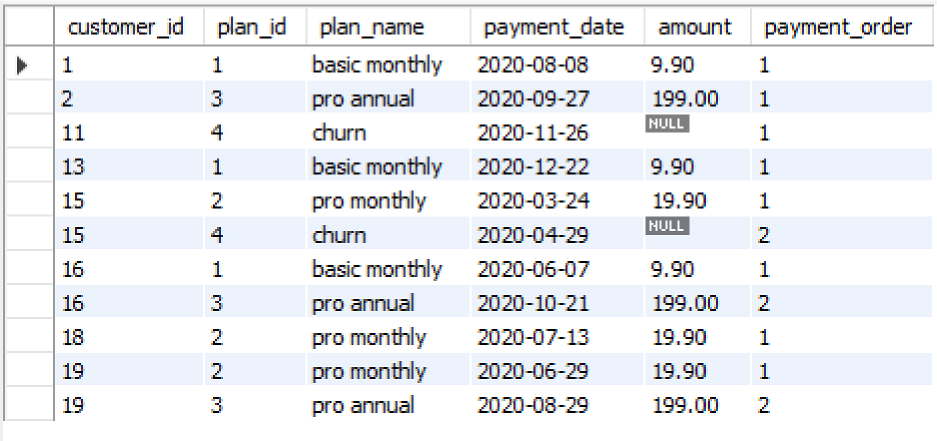
No Customer has downgraded to a basic monthly plan from a pro monthly plan since 2020.

C. Challenge Payment Question

The Foodie-Fi team wants you to create a new payments table for the year 2020 that includes amounts paid by each customer in the subscriptions table with the following requirements:

- Monthly payments always occur on the same day of the month as the original start_date of any monthly paid plan
- upgrades from basic to monthly or pro plans are reduced by the current paid amount in that month and start immediately
- Upgrades from pro monthly to pro annual are paid at the end of the current billing period and also start at the end of the month period
- Once a customer churns they will no longer make payments

```
# Creating table payments
CREATE  TABLE payments
SELECT
  customer_id,
  plan_id,
  plan_name,
  DATE(payment_date) AS payment_date,
  amount,
  RANK() OVER(
    PARTITION BY customer_id
    ORDER BY
      payment_date
  ) AS payment_order
FROM
  (
    SELECT
      customer_id,
      s.plan_id,
      plan_name,
      DATE_ADD(start_date, INTERVAL n.n MONTH) AS payment_date,
      CAST(price AS DECIMAL(5,2))  AS amount
    FROM subscriptions AS s
    JOIN plans AS p ON s.plan_id = p.plan_id
    JOIN (
        SELECT 0 AS n UNION ALL
        SELECT 1 AS n UNION ALL
        SELECT 2 AS n UNION ALL
        SELECT 3 AS n UNION ALL
        SELECT 4 AS n UNION ALL
        SELECT 5 AS n UNION ALL
        SELECT 6 AS n UNION ALL
        SELECT 7 AS n UNION ALL
        SELECT 8 AS n UNION ALL
        SELECT 9 AS n UNION ALL
        SELECT 10 AS n UNION ALL
        SELECT 11 AS n UNION ALL
        SELECT 12 AS n
      ) AS n
      ON n.n <= 12
    WHERE s.plan_id != 0 AND start_date < '2021-01-01'
    GROUP BY
      customer_id,
      s.plan_id,
      plan_name,
      start_date,
      price
  ) AS t
ORDER BY
  customer_id;
```



customer_id	plan_id	plan_name	payment_date	amount	payment_order
1	1	basic monthly	2020-08-08	9.90	1
2	3	pro annual	2020-09-27	199.00	1
11	4	churn	2020-11-26	NULL	1
13	1	basic monthly	2020-12-22	9.90	1
15	2	pro monthly	2020-03-24	19.90	1
15	4	churn	2020-04-29	NULL	2
16	1	basic monthly	2020-06-07	9.90	1
16	3	pro annual	2020-10-21	199.00	2
18	2	pro monthly	2020-07-13	19.90	1
19	2	pro monthly	2020-06-29	19.90	1
19	3	pro annual	2020-08-29	199.00	2

D. OUTSIDE-THE-BOX QUESTIONS

1. **How would you calculate the rate of growth for Foodie-Fi?**

   I. The current value subtracts the previous value, and then divides it by the previous value, multiplying by 100 to get the percentage result. If the value is greater than 0 then the growth is positive, if the value is below or equal to 0 then there is no growth.

   II. We can calculate **revenue growth** or **customer growth**, **year-over-year growth**, and **month-over-month growth.**

   III. Values need to be cleared before calculation, for example, if we calculate revenue we need to subtract refunds or chargebacks first as they are not in our revenue anymore.

   IV. For customers, it can be calculated as the growth of active customers (all customers subtracting churned customers and trial customers).

2. **What key metrics would you recommend Foodie-Fi management to track over time to assess the performance of their overall business?**

   I. **Monthly Recurring Revenue (MRR):** This metric measures the revenue generated from monthly subscription plans and can be used to track overall growth and predict future revenue.

   II. **Customer Acquisition Cost (CAC):** This metric measures the cost of acquiring new customers and can be used to assess the efficiency of marketing and sales efforts.

   III. **Customer Lifetime Value (CLV):** This metric measures the projected revenue a customer will generate over their lifetime and can be used to identify which customer segments are most valuable to the business.

   IV. **Retention Rate**: This metric measures the percentage of customers who continue to use the service after a certain period of time and can be used to assess the effectiveness of customer retention strategies.

   V. **Churn Rate:** This metric measures the percentage of customers who cancel their subscriptions and can be used to identify and address issues that may be causing customers to leave.

3. **What are some key customer journeys or experiences that you would analyze further to improve customer retention?**

    I. **Onboarding**: The process of welcoming new customers to the service and introducing them to the features and benefits of the service. Analyzing the onboarding process can help identify areas where the experience can be improved to better engage and retain new customers.

    II. **Payment and Billing:** The process of managing payments and billing for the service. Analyzing this process can help identify areas where the experience can be improved to reduce friction and make it easier for customers to pay and manage their subscriptions.

    III. **Customer Support:** The process of providing assistance to customers who need help using the service. Analyzing this process can help identify areas where the experience can be improved to provide better support and reduce customer frustration.

    IV. **Feature Adoption:** The process of customers discovering and using the features of the service. Analyzing this process can help identify areas where the experience can be improved to better educate customers on the value of the service and increase engagement with the service.

    V. **Cancellation:** The process of customers cancelling their subscriptions. Analyzing this process can help identify areas where the experience can be improved to reduce friction and make it easier for customers to cancel their subscriptions or identify why customers are cancelling and take actions to address the issues.

    VI. **Feedback**: The process of collecting and analyzing feedback from customers can provide valuable insights into what the customers like and dislike about the service, and identify areas for improvement.

4. **If the Foodie-Fi team were to create an exit survey shown to customers who wish to cancel their subscription, what questions would you include in the survey?**

   I. What was the primary reason for cancelling your subscription? (Multiple choice options such as price, lack of value, bad customer service, etc.)

   II. How satisfied were you with the service overall? (Scale of 1-10 or using a rating system like strongly disagree, disagree, neutral, agree, strongly agree)

   III. How satisfied were you with the customer service you received? (Scale of 1-10 or using a rating system like strongly disagree, disagree, neutral, agree, strongly agree)

   IV. How likely are you to recommend our service to a friend or family member? (Scale of 1-10 or using a rating system like strongly disagree, disagree, neutral, agree, strongly agree)

   V. How frequently did you use the service? (Multiple choice options such as, daily, weekly, monthly, rarely, etc.)

   VI. What features of the service did you find most valuable? (Free text)

   VII. What features of the service did you find least valuable? (Free text)

   VIII. Is there anything that the company could have done to keep you as a customer? (Free text)

   IX. Is there any other feedback you would like to provide? (Free text)

5. What business levers could the Foodie-Fi team use to reduce the customer churn rate? How would you validate the effectiveness of your ideas?

I. **Improving the customer onboarding experience:** By making the onboarding process more engaging and informative, the team can increase the chances of new customers becoming long-term users.

II. **Offering personalized pricing and plans:** By tailoring pricing and plans to individual customers' needs, the team can increase the perceived value of the service and reduce the likelihood of cancellation.

III. **Enhancing customer support:** By improving the quality of customer support, the team can reduce customer frustration and increase the chances of customers staying with the service.

IV. **Increasing engagement with the service:** By encouraging customers to use the service more frequently, the team can increase the perceived value of the service and reduce the likelihood of cancellation.

V. **Offering incentives for long-term commitment:** By offering incentives such as discounts for a long-term commitment, the team can increase the perceived value of the service and reduce the likelihood of cancellation.

VI. **Actively collecting and analyzing customer feedback:** By actively collecting and analyzing customer feedback, the team can identify areas for improvement in the service and take action to address customer complaints and dissatisfaction.

Links to all Necessary Files

1. Foodie-Fi Case Study: Link
2. Foodie-Fi Schema (SQL File): Link
3. A. Customer Journey (SQL File): Link
4. B. Data Analysis Questions (SQL File): Link
5. C. Payments Table (SQL File): Link