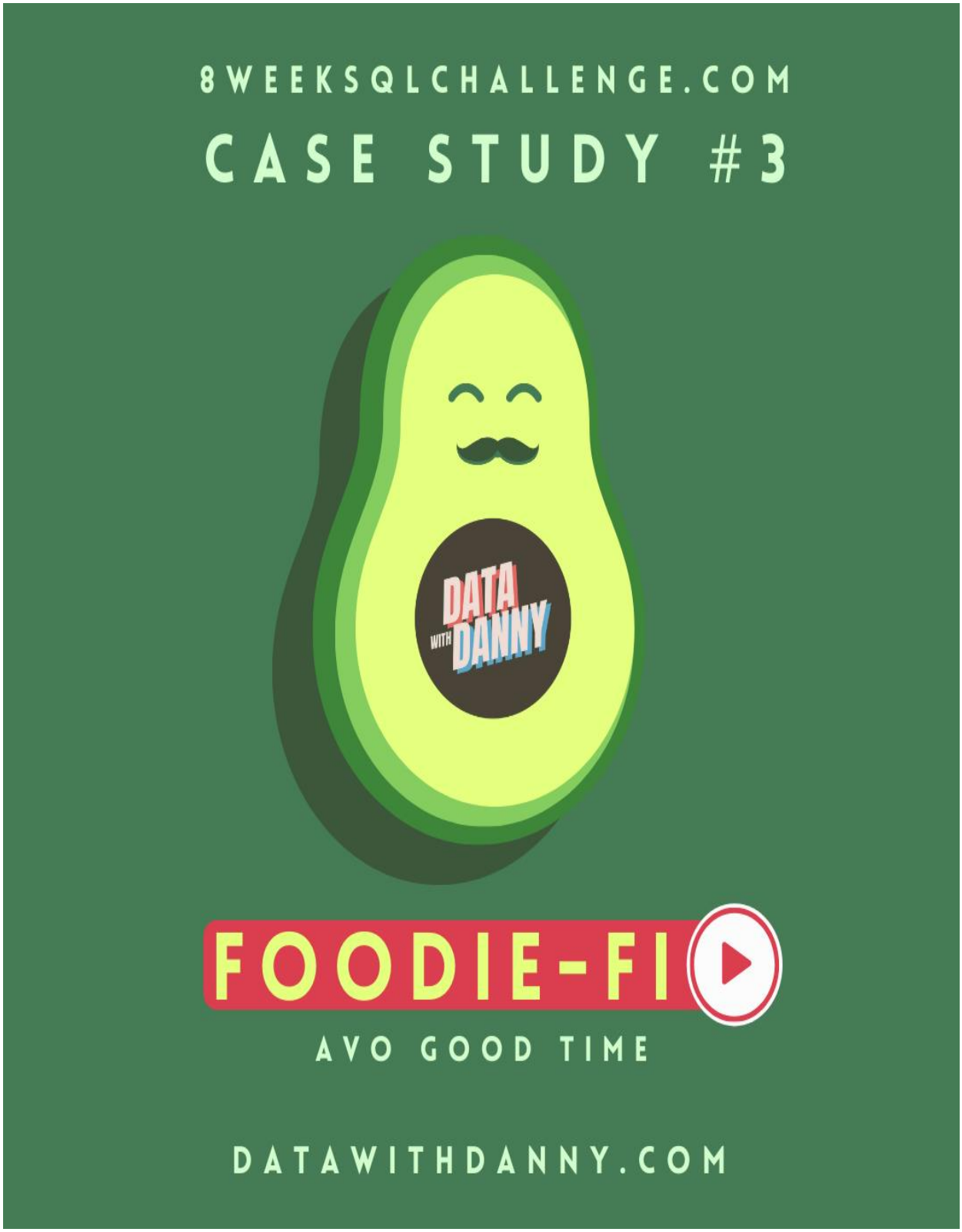
FOODIE-FI CASE STUDY ANALYSIS



PROJECT INTERN

NITHYA S

INTRODUCTION:

Danny realized that he wanted to create a new streaming service that only had foodrelated 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.

E-R DIAGRAM

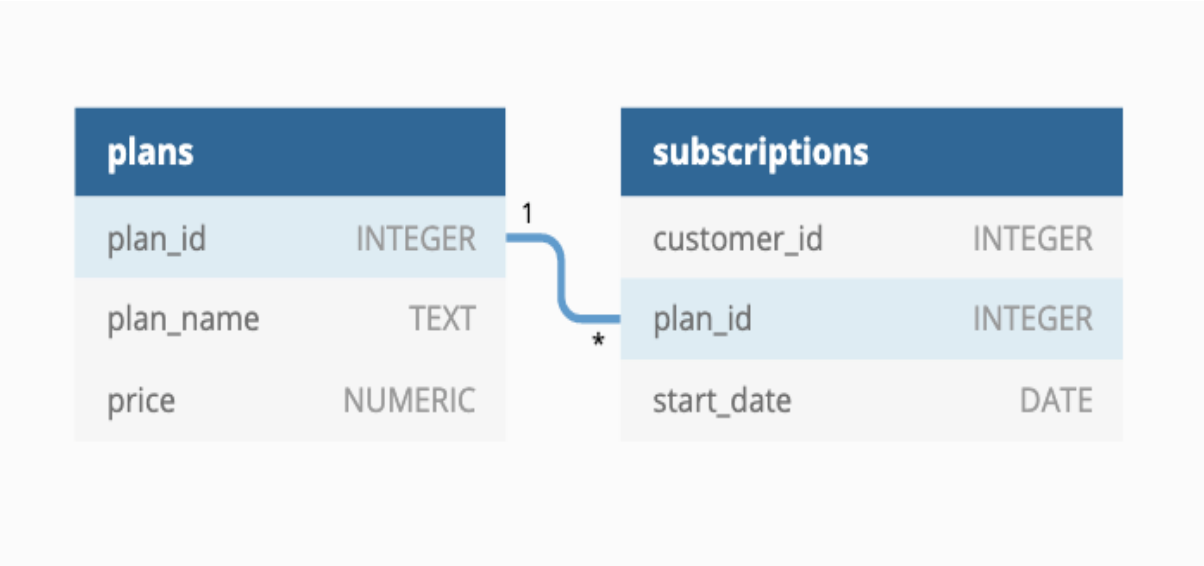


TABLE CREATION:

```
CREATE SCHEMA foodie_fi ;
USE foodie_fi;
CREATE TABLE plans(
plan_id SMALLINT unsigned,
plan_name VARCHAR(45) NOT NULL,
price DECIMAL,
PRIMARY KEY  (plan_id)
);
CREATE TABLE subscriptions(
customer_id SMALLINT unsigned,
plan_id SMALLINT unsigned,
```

start_date **DATE** NOT NULL ,

FOREIGN KEY (plan_id) REFERENCES plans (plan_id) ON DELETE CASCADE

);

TABLE 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 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

customer_id	plan_id	start_date
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

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

Try to keep it as short as possible - you may also want to run some sort of join to make your explanations a bit easier!

CUSTOMER1

customer_id	plan_id	start_date
1	0	2020-08-01
1	0	2020-08-01
1	1	2020-08-08

Customer 1 started the free trial on 1 Aug 2020 and subsequently subscribed to the basic monthly plan on 8 Aug 2020 after the 7-days trial has ended.

CUSTOMER2

Result Grid | Filter Rows:

customer_id	plan_id	start_date
2	0	2020-09-20
2	3	2020-09-27

Customer 2 started the free trial on 20 Sep 2020 and subsequently subscribed to the Pro-annually plan on 27 Sep 2020 after the 7-days trial has ended.

CUSTOMER11

Result Grid | Filter Rows:

customer_id	plan_id	start_date
11	0	2020-11-19
11	4	2020-11-26

Customer 11 started the free trial on 19 Nov 2020 and subsequently the customer **terminated** subscription and **churned** until the paid subscription ended on 19 Nov 2020

CUSTOMER13

Result Grid | Filter Rows:

customer_id	plan_id	start_date
13	0	2020-12-15
13	1	2020-12-22
13	2	2021-03-29

Customer 13 started the free trial on 15 Dec 2020 and subsequently subscribed to the basic-monthly plan on 22 Dec 2020 after the 7-days trial has ended and then subscribed to pro annual plan on Mar 29 2021.

CUSTOMER 15

customer_id	plan_id	start_date
15	0	2020-03-17
15	2	2020-03-24
15	4	2020-04-29

Customer 15 commenced **free trial** on 17 Mar 2020, then upgraded to **pro monthly** plan on 24 Mar 2020 after the trial ended.
In the following month on 29 Apr 2020, the customer **terminated** subscription and **churned** until the paid subscription ended on 24/25 May 2020.

CUSTOMER 16

customer_id	plan_id	start_date
16	0	2020-05-31
16	1	2020-06-07
16	3	2020-10-21

Customer 16 started the free trial on 31 May 2020 and subsequently subscribed to the basic-monthly plan on 07 Jun 2020 after the 7-days trial has ended and then subscribed to pro annual plan on Oct 10 2021.

CUSTOMER 18

customer_id	plan_id	start_date
18	0	2020-07-06
18	2	2020-07-13

Customer 18 started the free trial on 06 july 2020 and subsequently subscribed to the Pro-monthly plan on 13 July 2020 after the 7-days trial has ended.

CUSTOMER19

Result Grid | Filter Rows:

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

Customer 19 started the free trial on 22 July 2020 and subsequently subscribed to the pro-monthly plan on 29 July 2020 after the 7-days trial has ended and then subscribed to pro annual plan on Aug 29 2020.

Data Analysis Questions

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

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

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

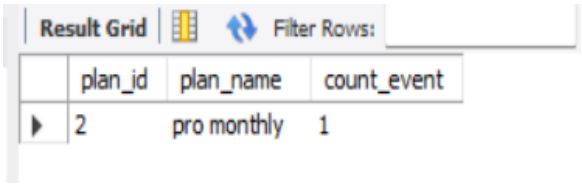
```
SELECT MONTH (start_date) AS MONTHS,
count(customer_id) AS NUM_CUSTOMERS  FROM SUBSCRIPTIONS group by MONTHS ;
```

MONTHS	NUM_CUSTOMERS
8	4
9	2
11	2
12	2
3	3
4	1
5	1
6	3
10	1
7	2

Aug has the biggest number of trial plan distribution.

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 P.plan_id, P.plan_name, COUNT(*) AS count_event FROM plans P JOIN subscriptions S
ON P.plan_id = S.plan_id
WHERE S.start_date > "2020-12-31" GROUP BY P.plan_id, P.plan_name ORDER BY P.plan_id;
```

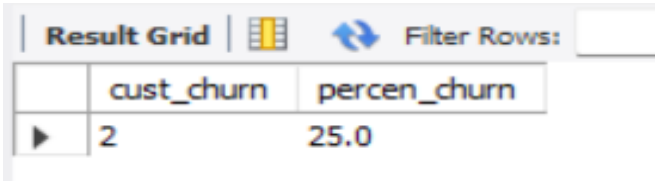


plan_id	plan_name	count_event
2	pro monthly	1

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 COUNT(*) AS cust_churn, ROUND(COUNT(*)*100/(SELECT COUNT(DISTINCT
customer_id) FROM subscriptions),1) AS percen_churn
FROM subscriptions WHERE plan_id = 4;
```



cust_churn	percen_churn
2	25.0

There are 2 customers who churned, which is 25.0 of 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 cte_churn AS (SELECT *, LAG(plan_id, 1) OVER(partition by customer_id order by plan_id)
AS prev_plan FROM subscriptions)
SELECT COUNT(prev_plan) AS cnt_churn, ROUND(COUNT(*)*100/(SELECT COUNT(DISTINCT
customer_id) FROM subscriptions),0) AS percn_churn FROM cte_churn
WHERE plan_id = 4 AND prev_plan=0;
```

	cnt_churn	percn_churn
▶	1	13

1 customers churned straight after the initial free trial which is 13% of entire customer base.

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

```
WITH cte_next_plan AS (SELECT *, LEAD(plan_id, 1) OVER(partition by customer_id order by plan_id) AS next_plan FROM subscriptions)
SELECT next_plan, COUNT(*) AS num_cust, ROUND(COUNT(*) * 100/(SELECT COUNT(DISTINCT customer_id) FROM subscriptions),1) AS perc_next_plan FROM cte_next_plan WHERE next_plan is not null AND plan_id = 0 group by next_plan order by next_plan;
```

	next_plan	num_cust	perc_next_plan
▶	0	1	12.5
	1	3	37.5
	2	3	37.5
	3	1	12.5
	4	1	12.5

More than 80% of customers are on paid plans with small 12.5% on plan 1 (pro annual $199). Foodie-Fi has to strategize on their customer acquisition who would be willing to spend more.

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

```
WITH cte_next_date AS(SELECT *, LEAD(start_date, 1) OVER(partition by customer_id order by start_date) AS next_date FROM subscriptions WHERE start_date <= "2020-12-31"), plans_breakdown AS(
SELECT plan_id, COUNT(DISTINCT customer_id) AS num_customer FROM cte_next_date WHERE (next_date is not null AND (start_date < "2020-12-31" AND next_date > "2020-12-31")) OR (next_date is null AND start_date <"2020-12-31") GROUP BY plan_id)
SELECT plan_id, num_customer, ROUND(num_customer * 100 / (SELECT COUNT(DISTINCT customer_id) FROM subscriptions), 1) AS perc_customer
FROM plans_breakdown GROUP BY plan_id, num_customer ORDER BY plan_id;
```

	plan_id	num_customer	perc_customer
▶	1	2	25.0
	2	1	12.5
	3	3	37.5
	4	2	25.0

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

```
SELECT COUNT(customer_id) FROM subscriptions WHERE plan_id = 3 AND start_date
 <= "2020-12-31";
```

	COUNT(customer_id)
▶	3

3 customers 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?

```
WITH annual_plan AS( SELECT customer_id, start_date AS Annual_date FROM subscriptions
WHERE plan_id = 3), trial_plan AS
(SELECT customer_id, start_date AS trial_date FROM subscriptions WHERE plan_id = 0) SELECT
ROUND(AVG(DATEDIFF(annual_date, trial_date)),0) AS avg_upgrade FROM annual_plan ap
JOIN trial_plan tp ON ap.customer_id = tp.customer_id;
```

	avg_upgrade
▶	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)

```
WITH annual_plan AS( SELECT customer_id, start_date AS Annual_date FROM subscriptions
WHERE plan_id = 3), trial_plan AS
(SELECT customer_id, start_date AS trial_date FROM subscriptions WHERE plan_id = 0), day_period
AS( SELECT DATEDIFF(annual_date, trial_date) AS diff FROM trial_plan tp
```

LEFT JOIN annual_plan ap **ON** tp.customer_id = ap.customer_id **WHERE** annual_date is not null),bins **AS**(**SELECT** *, FLOOR(diff/30) **AS** bins **FROM** day_period)

SELECT CONCAT((bins*30)+1, '-',(bins+1)*30,'days') **AS** days, **COUNT**(diff) **AS** total from bins group by bins;

	days	total
▶	1-30days	1
	121-150days	1
	61-90days	1

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

WITH next_plan **AS**(**SELECT** *, **LEAD**(plan_id,1) **OVER**(**partition by** customer_id ORDER **BY** start_date, plan_id) **AS** plan **FROM** subscriptions)

SELECT COUNT(distinct customer_id) **AS** num_downgrade **FROM** next_plan np **LEFT JOIN** plans p **ON** p.plan_id = np.plan_id

WHERE p.plan_name = "pro monthly" **AND** np.plan = 1 **AND** start_date <= "2020-12-31";

	num_downgrade
▶	0

No customer has downgrade from pro monthly to basic monthly in 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:

create view customer_id1 **as** (
(**select** customer_id, plan_id, plan_name, start_date **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id =1) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 1 month) **as** "payment_date", price **as** "amount" **from** plans join subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 1) **union** all

(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 2 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 1) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 3 month) **as** "payment_date", price **as** "amount" from plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 1) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 4 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 1));

create view customer_id2 as (
(**select** customer_id, plan_id, plan_name, start_date **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 2));

create view customer_id13 as (
(**select** customer_id, plan_id, plan_name, start_date **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using (plan_id) **where** plan_id not in (0,4) **and** customer_id = 13 limit 1));

create view customer_id15 as (
(**select** customer_id, plan_id, plan_name, start_date as "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 15) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 1 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 15));

create view customer_id16 as (
(**select** customer_id, plan_id, plan_name, start_date **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 16 limit 1) union all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 1 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 16 limit 1) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 2 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 16 limit 1) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 3 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id not in (0,4) **and** customer_id = 16 limit 1) **union** all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 4 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) where plan_id not in (0,4) **and** customer_id = 16 limit 1) union all
(**select** customer_id, plan_id, plan_name, date_add(start_date, interval 5 month) **as** "payment_date", price **as** "amount" **from** plans **join** subscriptions using(plan_id) **where** plan_id = 3 **and** customer_id = 16));

```
create view customer_id18 as (
(select customer_id, plan_id, plan_name, start_date as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id = 2 and customer_id = 18) union all
(select customer_id, plan_id, plan_name, date_add(start_date, interval 1 month) as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id = 2 and customer_id = 18) union all
(select customer_id, plan_id, plan_name, date_add(start_date, interval 2 month) as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id = 2 and customer_id = 18) union all
(select customer_id, plan_id, plan_name, date_add(start_date, interval 3 month) as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id = 2 and customer_id = 18) union all
(select customer_id, plan_id, plan_name, date_add(start_date, interval 4 month) as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id = 2 and customer_id = 18) union all
(select customer_id, plan_id, plan_name, date_add(start_date, interval 5 month) as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id = 2 and customer_id = 18));


create view customer_id19 as (
(select customer_id, plan_id, plan_name, start_date as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id not in (0,4) and customer_id = 19 limit 1) union all
(select customer_id, plan_id, plan_name, date_add(start_date, interval 1 month) as "payment_date", price as "amount" from plans join subscriptions using(plan_id) where plan_id not in (0,4) and customer_id = 19)) ;


create table payments as (
select *, rank() over(order by payment_date) as "payment_order" from customer_id1 union all
select *, rank() over(order by payment_date) as "payment_order" from customer_id2 union all
select *, rank() over(order by payment_date) as "payment_order" from customer_id13 union all
select *, rank() over(order by payment_date) as "payment_order" from customer_id15 union all
select *, rank() over(order by payment_date) as "payment_order" from customer_id16 union all
select *, rank() over(order by payment_date) as "payment_order" from customer_id18 union all
select *, rank() over(order by payment_date) as "payment_order" from customer_id19);

select * from payments;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	plan_id	plan_name	payment_date	amount	payment_order
1	1	basic monthly	2020-08-08	9.90	1
1	1	pro monthly	2020-09-08	9.90	2
1	1	pro monthly	2020-10-08	9.90	3
1	1	pro monthly	2020-11-08	9.90	4
1	1	pro monthly	pro monthly	9.90	5
2	3	pro annual	2020-09-27	199.00	1
13	1	basic monthly	2020-12-22	9.90	1
15	2	pro monthly	2020-03-24	19.90	1
15	2	pro monthly	2020-04-24	9.90	2
16	1	basic monthly	2020-06-07	9.90	1
16	1	basic monthly	2020-07-07	9.90	2
16	1	basic monthly	2020-08-07	9.90	3
16	1	basic monthly	2020-09-07	9.90	4
16	3	pro annual	2020-10-21	189.10	6
18	2	pro monthly	2020-07-13	19.90	1
18	2	pro monthly	2020-08-13	19.90	2

D.OUTSIDE-THE-BOX QUESTIONS

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

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

- **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.
- **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.

- **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.
- **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.
- **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?

- **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.
- **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.
- **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.
- **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.
- **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?

- What was the primary reason for cancelling your subscription? (Multiple choice options such as price, lack of value, bad customer service, etc.)
- 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)
- 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)
- 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)
- How frequently did you use the service? (Multiple choice options such as, daily, weekly, monthly, rarely, etc.)
- 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)
- Is there anything that the company could have done to keep you as a customer? (Free text)
- 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?

- **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.
- **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.
- **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.
- **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.

- **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.
- **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.