

# Case Study #3 - Foodie-Fi

Danny Ma · May 18, 2021

-Dhiraj Kumar Choudhary



## 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!

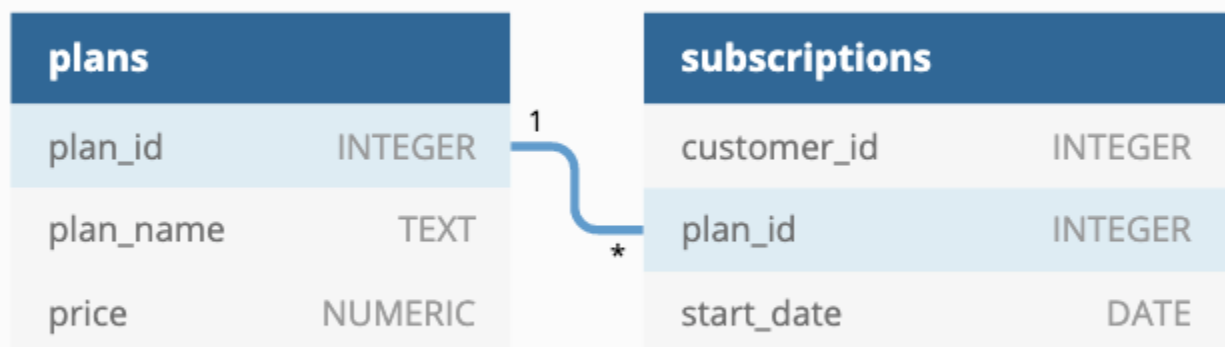
Danny created Foodie-Fi with a data driven mindset and wanted to ensure all future investment decisions and new features were decided using data. This case study focuses on using subscription style digital data to answer important business questions.

## Available Data

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.

All datasets exist within the `foodie-fi` database schema - be sure to include this reference within your SQL scripts as you start exploring the data and answering the case study questions.

## Entity Relationship Diagram



**Table 1: plans**

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

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.

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.

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

**Table 2: subscriptions**

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 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
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

## Interactive SQL Instance

You can use the embedded DB Fiddle below to easily access these example datasets - this interactive session has everything you need to start solving these questions using SQL.

You can click on the [Edit on DB Fiddle](#) link on the top right hand corner of the embedded session below and it will take you to a fully functional SQL editor where you can write your own queries to analyse the data.

You can feel free to choose any SQL dialect you'd like to use, the existing Fiddle is using PostgreSQL 13 as default.

Serious SQL students will have access to the same relevant schema SQL and example solutions which they can use with their Docker setup from within the course player!

## Case Study Questions

This case study is split into an initial data understanding question before diving straight into data analysis questions before finishing with 1 single extension challenge.

### A. Customer Journey

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!



**SQL Query:**


```

8
9 • SELECT
10     s.customer_id, p.plan_name, s.start_date
11 FROM
12     subscriptions s
13     JOIN
14     plans p ON s.plan_id = p.plan_id;
15

```

### Output Table:

Result Grid     Filter Rows: <input type="text"/>			
	customer_id	plan_name	start_date
▶	1	trail	2020-08-01
	2	trail	2020-09-20
	11	trail	2020-11-19
	13	trail	2020-12-15
	15	trail	2020-03-17
	16	trail	2020-05-31
	18	trail	2020-07-06
	19	trail	2020-06-22
	1	basic trail	2020-08-08
	13	basic trail	2020-12-22
	16	basic trail	2020-06-07
	13	pro monthly	2021-03-29
	15	pro monthly	2020-03-24
	18	pro monthly	2020-07-13
	19	pro monthly	2020-06-29
	2	pro annual	2020-09-27

Result 1 x 

since there are 8 Customer here is the detailed customer's journey.

- Customer\_id 1 -> trail -> 2020-08-01 -> upgrade -> basic trail
- Customer\_id 2 -> trail -> 2020-09-20 -> upgrade -> pro annual
- Customer\_id 11 -> trail -> 2020-11-19 -> upgrade -> churan
- Customer\_id 13 -> trail -> 2020-12-15 -> upgrade -> basic trail -> upgrade -> pro monthly
- Customer\_id 15 -> trail -> 2020-03-07 -> upgrade -> pro monthly -> churan
- Customer\_id 16 -> trail -> 2020-05-31 -> upgrade -> basic trail -> pro annual
- Customer\_id 18 -> trail -> 2020-07-06 -> upgrade -> pro monthly
- Customer\_id 19 -> trail -> 2020-06-22 -> upgrade -> pro monthly -> pro annual

## B. Data Analysis Questions

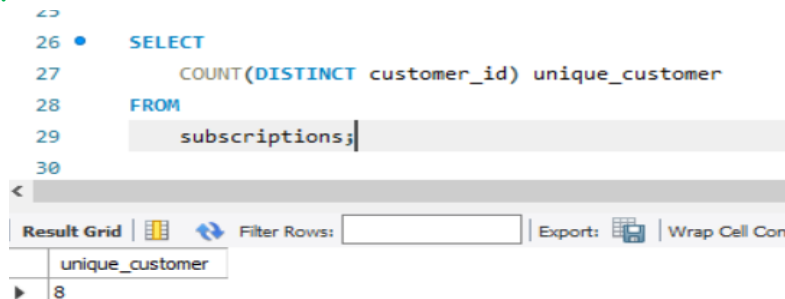
1. How many customers has Foodie-Fi ever had?

Ans:

```

26 • SELECT
27     COUNT(DISTINCT customer_id) unique_customer
28 FROM
29     subscriptions;
30

```



unique_customer
8

It was found that Foodie-Fi has 8 customer.

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

Ans:

```

35
36 • SELECT
37     MONTH(start_date) AS months,
38     COUNT(customer_id) AS number_of_customers
39 FROM
40     subscriptions
41 GROUP BY months;
42

```

months	number_of_customers
8	3
9	2
11	2
12	2
3	3
4	1
5	1
6	3
10	1
7	2

- It was found that in month of March, June, August has the highest number of trail 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`

Ans:

```

45 -- 3
46 • SELECT
47     p.plan_name, p.plan_id, COUNT(*) AS count_of_events
48 FROM
49     subscriptions s
50 JOIN
51     plans p ON p.plan_id = s.plan_id
52 WHERE
53     s.start_date >= '2021-01-01'
54 GROUP BY p.plan_id, p.plan_name
55 ORDER BY p.plan_id;
56
57

```

plan_name	plan_id	count_of_events
pro monthly	2	1

- It was found pro monthly plan `start_date` values occur after the year 2020 after the our dataset.
- since pro monthly is the plan 2 and it has been occurred only once via `customer_id` 13.



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

Ans:

```
59 -- 4
60
61 • SELECT
62     COUNT(*) AS customer_count,
63     ROUND(COUNT(*) * 100 / (SELECT
64         COUNT(DISTINCT customer_id)
65     FROM
66         subscriptions),
67     1) AS percentage_customer_churn
68 FROM
69     subscriptions
70 WHERE
71     plan_id = 4;
72
```

Result Grid

customer_count	percentage_customer_churn
2	25.0

- It was observed that there are 2 customer who have percentage of customer
- who have churned the round place to 1 decimal place is 25.0

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

Ans:

```
75 -- 5
76
77
78 • with base as(
79     select customer_id, start_date, plan_id, row_number() over(partition by customer_id order by plan_id asc) as rank_plan
80     from foodie-fi.subscriptions)
81 select
82     count(case when plan_id = 4 then 1 end) as free_trial_churn,
83     round(100*(count(case when plan_id = 4 then 1 end)/count(*)),2) as churn_percentage
84 from base where rank_plan = 2;
85
```

Result Grid

free_trial_churn	churn_percentage
1	12.50

- It was observed that one customers have churned straight after their initial free trial and 12.5 percentage.

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

Ans:

```
90 WITH next_plan_cte AS (  
91     SELECT  
92         customer_id,  
93         plan_id,  
94         LEAD(plan_id, 1) OVER(PARTITION BY customer_id ORDER BY plan_id) AS next_plan  
95     FROM foodie-fi.subscriptions)  
96  
97     SELECT  
98         next_plan,  
99         COUNT(*) AS conversions,  
100        ROUND(100 * COUNT(*) / (  
101            SELECT COUNT(DISTINCT customer_id)  
102            FROM foodie-fi.subscriptions),1) AS conversion_percentage  
103     FROM next_plan_cte  
104     WHERE next_plan IS NOT NULL  
105           AND plan_id = 0  
106     GROUP BY next_plan  
107     ORDER BY next_plan;
```

Result Grid

	next_plan	conversions	conversion_percentage
1	1	3	37.5
2	2	3	37.5
3	3	1	12.5
4	4	1	12.5

We got,

- plan1 -> 3 conversion -> 37.5%
- plan2 -> 3 conversion -> 37.5%
- plan3 -> 1 conversion -> 12.5%
- plan4 -> 1 conversion -> 12.5%

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

Ans:

```

113 WITH next_plan AS (
114     SELECT
115         customer_id,
116         plan_id,
117         start_date,
118         LEAD(start_date, 1) OVER (PARTITION BY customer_id ORDER BY start_date) AS next_date
119     FROM foodie-fi.subscriptions
120     WHERE start_date <= '2020-12-31'
121 ),
122 customer_breakdown AS (
123     SELECT plan_id, COUNT(DISTINCT customer_id) AS customers
124     FROM next_plan
125     WHERE (next_date IS NOT NULL AND (start_date < '2020-12-31' AND next_date > '2020-12-31'))
126        OR (next_date IS NULL AND start_date < '2020-12-31')
127     GROUP BY plan_id)
128
129 SELECT plan_id, customers,
130        ROUND(100 * customers / (
131            SELECT COUNT(DISTINCT customer_id)
132            FROM foodie-fi.subscriptions),1) AS percentage
133 FROM customer_breakdown
134 GROUP BY plan_id, customers
135 ORDER BY plan_id;

```

plan_id	customers	percentage
1	2	25.0
2	1	12.5
3	3	37.5
4	2	25.0

- It is observed that 3 customers have upgraded to the pro annual pack with 37.5%

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

Ans:

```

160
161 -- 8
162
163 • SELECT
164     COUNT(DISTINCT customer_id) AS unique_customer
165 FROM
166     foodie-fi.subscriptions
167 WHERE
168     plan_id = 3
169     AND start_date <= '2020-12-31';
170

```

unique_customer
3

- It was found that 3 customer have upgraded to annual pack.

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

Ans:

```
173 -- 9
174 WITH
175     trial_plan AS
176     (SELECT
182         annual_plan AS
183         (SELECT
184             customer_id,
185             start_date AS annual_date
186         FROM subscriptions
187         WHERE plan_id = 3
188         )
189
190     SELECT
191         ROUND(AVG(annual_date - trial_date),0) AS avg_days_to_upgrade
192     FROM trial_plan tp
193     JOIN annual_plan ap
194         ON tp.customer_id = ap.customer_id;
195
```

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

avg_days_to_upgrade
235

- It was found 235 days on average does it take for a customer 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).

Ans:

```
217 with base as (
218     select
219         customer_id,
220         plan_id,
221         start_date,
222         (lead(start_date) over(partition by customer_id order by start_date asc)) as lead_start_date
223     from Foodie-Fi.subscriptions
224     where plan_id in(0,3)),
225
226     tb as (select *, (lead_start_date - start_date) as diff from base where lead_start_date is not null),
227
228     tbl as (select
229         (case
244             count(*) as customers
245         from tb
246         group by month_interval
247         order by month_interval)
248
249     select breakdown_period, customers
250     from tbl
```

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

breakdown_period	customers
0 - 30 days	1
180 - 210 days	1

- It was observed that there is 1 customer in the breakdown period of 0-30 days & again 1 customer in 180-210 days.

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

Ans:

```
258 with base as (select
259     customer_id,
260     plan_id,
261     start_date,
262     (lead(plan_id) over(partition by customer_id order by start_date asc)) as lead_plan_id,
263     (lead(start_date) over(partition by customer_id order by start_date asc)) as lead_start_date
264 from foodie-fi.subscriptions
265 where extract(year from start_date) = 2020)
266
267 select count(*) as downgraded
268 from base
269 where lead_plan_id is not null and
270        lead_plan_id = 2
271        and plan_id = 1;
272
```

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

downgraded
0

- It was found that none of the customer have downgraded the plan.

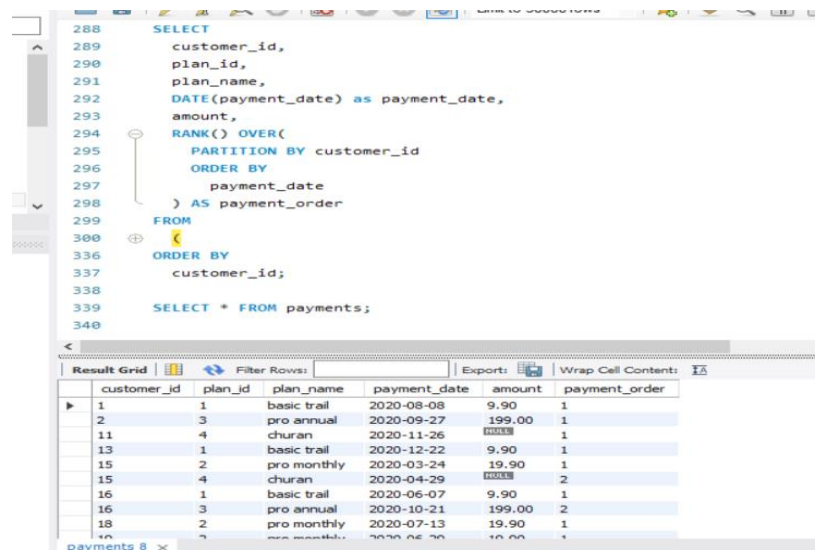
### C. Challenge Payment Question

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 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 starts at the end of the month period
- once a customer churns they will no longer make payments

Example outputs for this table might look like the following:

## SQL Query:



The screenshot shows a SQL IDE with a query editor and a results grid. The query is as follows:

```
288 SELECT
289     customer_id,
290     plan_id,
291     plan_name,
292     DATE(payment_date) as payment_date,
293     amount,
294     RANK() OVER(
295         PARTITION BY customer_id
296         ORDER BY
297             payment_date
298     ) AS payment_order
299 FROM
300     payments
301 ORDER BY
302     customer_id;
303
304 SELECT * FROM payments;
```

The results grid displays the following data:

	customer_id	plan_id	plan_name	payment_date	amount	payment_order
1	1	1	basic trail	2020-08-08	9.90	1
2	3	3	pro annual	2020-09-27	199.00	1
11	4	4	churan	2020-11-26	19.90	1
13	1	1	basic trail	2020-12-22	9.90	1
15	2	2	pro monthly	2020-03-24	19.90	1
15	4	4	churan	2020-04-29	19.90	2
16	1	1	basic trail	2020-06-07	9.90	1
16	3	3	pro annual	2020-10-21	199.00	2
18	2	2	pro monthly	2020-07-13	19.90	1
19	3	3	pro monthly	2020-06-20	19.90	1

## D. Outside The Box Questions

The following are open ended questions which might be asked during a technical interview for this case study - there are no right or wrong answers, but answers that make sense from both a technical and a business perspective make an amazing impression!

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

Ans:

**Customer growth:**  $(\text{new\_total} - \text{old\_total}) / \text{old\_total}$  for active subscriptions, **Financial results:** Total income and profit compared to last period analyzed, and also break those numbers through plan types.

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

Ans: Revenue, Retention Rate, Number of paying users, monthly revenue, Churn rate, Customer review.

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

Ans: Providing exciting offers, cold talks with customers.

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?

Ans: Few of the question might be like given below,

- How often the customer used the service?
- how likely is it that you would recommend Foodie-Fi services to your friends?
- Did you find any content that you liked? (Yes/No)
- Reason for leaving in term of feedback.

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?

Ans: We might reduce the retention rate by adopting the following measures,

- 1)Solve problems when they face,
- 2)Give attention to the most valuable customers,
- 3)Remind customers of the value you provide.

It would also be great to create a community around customers.

Most common way is - Collaborate with Content Creators: engage with content creators on social

media to create a meaningful connections with users and also to gain recognition for the brand, start challenges, drop special content. Measure with % of new subscribers that stayed

## **Conclusion**

This case study should reflect realistic questions we usually focus on for all product related analytics requests in a wide variety of industries, especially in the digital space!

Here I have tried my best to provide the best resolution on the case study of foodie-fie.

Link: [ProjectFoodFie](#)

## **Thank You**