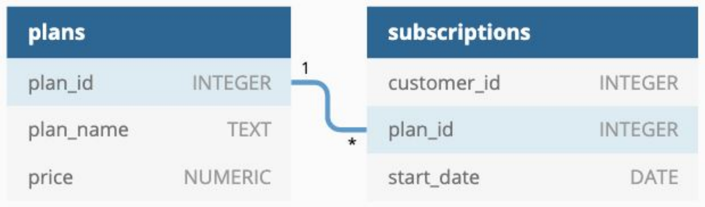
[SQL] Customer Subscription for Streaming Services

작성자 : 김유민

Introduction

Foodie-Fi, a subscription streaming service for food-related content, wants to utilize data to better understand customer retention, churn, and growth.

Entity Relationship Diagram



Datasets used

Table 1: `plans`

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

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

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.

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

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.

- Query :

```
DROP TABLE IF EXISTS subs_plans;
CREATE TABLE subs_plans AS (
  SELECT s.customer_id,
    s.plan_id,
    p.plan_name,
    p.price,
    s.start_date
  FROM foodie_fi.subscriptions AS s
    JOIN foodie_fi.plans AS p ON s.plan_id = p.plan_id
);
```

```
SELECT customer_id,
  plan_name,
  start_date
FROM foodie_fi.subs_plans
WHERE customer_id IN (1, 2, 11, 13, 15, 16, 18, 19)
ORDER BY customer_id,
  plan_id ASC;
```

- Results :

customer_id	plan_name	start_date
1	trial	2020-08-01
1	basic monthly	2020-08-08
2	trial	2020-09-20
2	pro annual	2020-09-27
11	trial	2020-11-19
11	churn	2020-11-26

customer_id	plan_name	start_date
13	trial	2020-12-15
13	basic monthly	2020-12-22
13	pro monthly	2021-03-29
15	trial	2020-03-17
15	pro monthly	2020-03-24
15	churn	2020-04-29
16	trial	2020-05-31
16	basic monthly	2020-06-07
16	pro annual	2020-10-21
18	trial	2020-07-06
18	pro monthly	2020-07-13
19	trial	2020-06-22
19	pro monthly	2020-06-29
19	pro annual	2020-08-29

- **Client #1**: upgraded to the basic monthly subscription within their 7 day trial period.
- **Client #2**: upgraded to the pro annual subscription within their 7 day trial period.
- **Client #11**: cancelled their subscription within their 7 day trial period.
- **Client #13**: upgraded to the basic monthly subscription within their 7 day trial period and upgraded to pro annual 3 months later.
- **Client #15**: upgraded to the pro annual subscription within their 7 day trial period and cancelled the following month.
- **Client #16**: upgraded to the basic monthly subscription after their 7 day trial period and upgraded to pro annual almost 5 months later.
- **Client #18**: upgraded to the pro monthly subscription within their 7 day trial period.
- **Client #19**: upgraded to the pro monthly subscription within their 7 day trial period and upgraded to pro annual 2 months later.

Data Analyzing - Inflow/Retention/Churn Metrics

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

- Query :

```
SELECT count(DISTINCT customer_id) AS 'n_customer'
FROM foodie_fi.subscriptions;
```

- Results : Total 1000 customer has subscribed.

n_customer
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

- Query :

```
SELECT month(start_date) as 'month',
       count(DISTINCT customer_id) as 'n_customers'
FROM foodie_fi.subs_plans
WHERE plan_id=0
GROUP BY month(start_date);
```

- Results : The largest number of customers flowed in in March, and the lowest number of subscribers in February.

month	n_customers
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

- Query :

```
SELECT plan_id,plan_name, count(plan_name) AS n_plans
FROM foodie_fi.sub_plans
WHERE start_date >= '2021-01-01'
GROUP BY plan_id,plan_name
ORDER BY plan_id;
```

- Results : Since 2020, there have been the largest number of customers who have churned, but there have also been many customers flowing into ‘pro-annual plan’.

plan_id	plan_name	n_plans
1	basic monthly	8
2	pro monthly	60
3	pro annual	63
4	churn	71

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

- Query :

```
SELECT plan_name, count(DISTINCT customer_id) as 'churned customers' , ROUND(100 * count(DISTINCT customer_id) /
             (SELECT count(DISTINCT customer_id)
              FROM foodie_fi.sub_plans ),2) AS 'percentage'
FROM foodie_fi.sub_plans
GROUP BY plan_name
HAVING plan_name = 'churn'
```

- Results : 307 customers have churned.

plan_name	churned customers	percentage
churn	307	30.70

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

- Query :

```
WITH next_plan_cte AS -- create cte1
    (SELECT *,
          lead(plan_id, 1) over(PARTITION BY customer_id
                                ORDER BY start_date) AS next_plan
     FROM foodie_fi.subscriptions),
     churners AS -- create cte2
    (SELECT *
     FROM next_plan_cte
     WHERE next_plan=4
       AND plan_id=0)

SELECT count(customer_id) AS 'churn after trial',
       round(100 *count(customer_id)/
               (SELECT count(DISTINCT customer_id) AS 'distinct customers'
                FROM foodie_fi.subscriptions), 2) AS 'percentage'
FROM churners;
```

- Results : 9.2% of customers have churned straight after their initial free trial. And it also means 90.8% of customers successfully converged into premium plan.

churn after trial	percentage
92	9.2

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

- Query :

```
WITH table1 AS -- create cte1
    (SELECT *,
          lead(plan_id, 1) over(PARTITION BY customer_id
                                ORDER BY start_date) AS next_plan
     FROM foodie_fi.sub_plans),

     table2 AS -- create cte2
    (SELECT *
     FROM table1
     WHERE next_plan IS NOT NULL
       AND plan_id=0)

SELECT next_plan, p.plan_name, count(next_plan) as 'plan after trial',
       round(100* count(next_plan)/
       (select count(distinct customer_id) from foodie_fi.sub_plans),2) as 'percentage'
FROM table2 as t
     INNER JOIN foodie_fi.plans as p ON t.next_plan = p.plan_id
GROUP BY next_plan,p.plan_name
```

- Results : They converged into basic monthly > pro monthly > pro annual plan after first time to trial the platform.

next_plan	plan_name	plan after trial	percentage
1	basic monthly	546	54.60
2	pro monthly	325	32.50
3	pro annual	37	3.70
4	churn	92	9.20

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

- Query :

```
WITH latest_plan_cte AS
  (SELECT *,
          row_number() over(PARTITION BY customer_id
                            ORDER BY start_date DESC) AS latest_plan
   FROM foodie_fi.sub_plans
   WHERE start_date <='2020-12-31' )

SELECT plan_id,
       plan_name,
       count(customer_id) AS customer_count,
       round(100*count(customer_id) /
               (SELECT COUNT(DISTINCT customer_id)
                FROM foodie_fi.subscriptions), 2) AS percentage_breakdown
FROM latest_plan_cte
WHERE latest_plan = 1
GROUP BY plan_id,plan_name
ORDER BY plan_id;
```

- Results : Until 2020, consumers mostly subscribed pro-monthly.

plan_id	plan_name	customer_count	percentage_breakdown

0	trial	19	1.90
1	basic monthly	224	22.40
2	pro monthly	326	32.60
3	pro annual	195	19.50
4	churn	236	23.60

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

- Query :

```
WITH previous_plan_cte AS
  (SELECT *,
          lag(plan_id, 1) over(PARTITION BY customer_id
                               ORDER BY start_date) AS previous_plan_id
   FROM foodie_fi.sub_plans)

SELECT count(customer_id) as 'count'
FROM previous_plan_cte
WHERE previous_plan_id<3
  AND plan_id=3
  AND year(start_date) = 2020;
```

- Results : we can know the number of customer who upgraded the plan at certain year.

count
195

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

- Query :

```
WITH first_plan_customer_cte AS
  (SELECT *,
          row_number() over(PARTITION BY customer_id
                            ORDER BY start_date ASC) AS plan_order
   FROM foodie_fi.sub_plans),

     annual_plan_customer_cte AS
  (SELECT *
   FROM foodie_fi.sub_plans
   WHERE plan_id=3)

SELECT round(avg(datediff(annual_plan_customer_cte.start_date, first_plan_customer_cte.start_date)), 2)AS avg_conversion_days
FROM first_plan_customer_cte
INNER JOIN annual_plan_customer_cte USING (customer_id)
WHERE plan_order = 1
```

- Results : Among the customer who converged into annual plan, they takes 104.62 days.

avg_days
104.62

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

- Query :

```
WITH first_plan_customer_cte AS
  (SELECT *,
          row_number() over(PARTITION BY customer_id
                            ORDER BY start_date ASC) AS  plan_order
   FROM foodie_fi.sub_plans),

     annual_plan_customer_cte AS
  (SELECT *
   FROM foodie_fi.sub_plans
   WHERE plan_id=3),

     diff_plan_cte AS
  ( SELECT f.customer_id, f.start_date as first_date, a.start_date as upgrade_date, round(datediff(a.start_date, f.start_date) / 3
    FROM first_plan_customer_cte AS f
    INNER JOIN annual_plan_customer_cte AS a ON f.customer_id = a.customer_id
    WHERE plan_order = 1)
```

```
SELECT CASE
    WHEN date_periods = 0 THEN '0 - 30days'
    ELSE CONCAT((date_periods * 30 + 1), ' - ' , (date_periods+1) * 30, ' days')
      END AS time_period, count(customer_id) AS customer_count
from diff_plan_cte
GROUP BY time_period ;
```

- Results : we can narrow down the average value (=104.62 days) into certain periods

time_period	customer_count
0 - 30days	41
151 - 180 days	41
61 - 90 days	30
31 - 60 days	25
121 - 150 days	33
91 - 120 days	39
181 - 210 days	32
211 - 240 days	7
361 - 390 days	1
241 - 270 days	4
271 - 300 days	3
301 - 330 days	1
331 - 360 days	1

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

- Query :

```
WITH next_plan_cte AS
  (SELECT *,
          lead(plan_id, 1) over(PARTITION BY customer_id
                                ORDER BY start_date) AS next_plan
   FROM foodie_fi.subscriptions)

SELECT count(*) AS downgrade_count
FROM next_plan_cte
WHERE plan_id=2
  AND next_plan=1
  AND year(start_date);
```

- Results : Hopefully, there is no one who downgraded the plan.

downgrade_count
0