



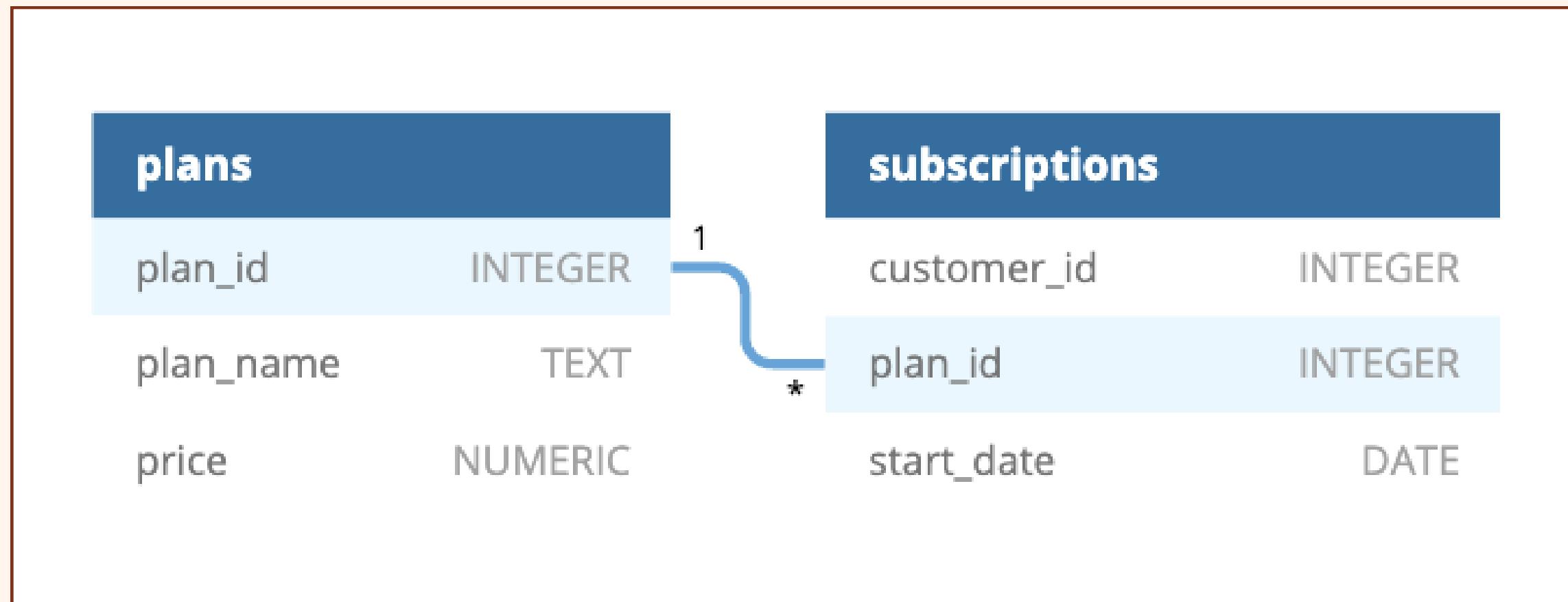
SQL PROJECT ON SUBSCRIPTION BASED BUSINESSES

Zahrah Aulia Putri



DATA

- The dataset was obtained from [8weeksqlchallenge](#).
- There are two tables, namely plans and subscriptions with the following ERD



- Data analysis was performed using **SQLite**.



Descriptions for Each Tables

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.





Descriptions for Each Tables

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 Journey

Based off the 3 sample customers provided in the sample from the subscriptions table, write a brief description about each customer's onboarding journey.

```
SELECT
    s.customer_id,
    s.plan_id,
    p.plan_name,
    p.price,
    s.start_date
FROM
    subscriptions AS s
INNER JOIN
    plans AS p
ON p.plan_id = s.plan_id
WHERE
    s.customer_id IN (2,4,7)
ORDER BY
    s.customer_id
```

	customer_id	plan_id	plan_name	price	start_date
1		2	0 trial	0	2020-09-20
2		2	3 pro annual	199	2020-09-27
3		4	0 trial	0	2020-01-17
4		4	1 basic monthly	9.9	2020-01-24
5		4	4 churn	NULL	2020-04-21
6		7	0 trial	0	2020-02-05
7		7	1 basic monthly	9.9	2020-02-12
8		7	2 pro monthly	19.9	2020-05-22

- **Customer 2** started initial 7 days free trial at 2020-09-20. During the trial, Customer 2 **upgrade** to pro annual plan and the pro annual plan started on 2020-09-27 right after the free trial finished.
- **Customer 4** started initial 7 days free trial at 2020-01-17. During the trial, Customer 4 **downgrade** to basic plan and the basic plan started on 2020-01-24 right after the free trial finished. After **almost 3 months** using the service, Customer 4 decided to **cancel** their service on 2020-04-21 but the customer still had the access until 2020-04-24.
- **Customer 7** started initial 7 days free trial at 2020-02-05. During the trial, Customer 7 **downgrade** to basic plan and the basic plan started on 2020-02-12 right after the free trial finished. After **more than 3 months** using the basic plan, Customer 7 decided to **upgrade** to pro monthly plan and the plan effect straightaway on 2020-05-22.





Data Analysis

To make analysis more efficient, I made a View named V_SubsPlan with query below.

```
- CREATE VIEW V_SubsPlan AS
  SELECT
    s.customer_id,
    s.plan_id,
    p.plan_name,
    p.price,
    s.start_date
  FROM
    subscriptions AS s
  INNER JOIN
    plans AS p
  ON p.plan_id = s.plan_id
  ORDER BY
    s.customer_id
```



Data Analysis

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

```
SELECT  
    count(DISTINCT  
        s.customer_id) AS [Total Customers]  
FROM  
    subscriptions AS s
```

Total Customers	
1	1000

There are **1.000 total customers** that Foodie-Fie has ever had.





Data Analysis

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
    strftime('%Y-%m-01', start_date) AS First_Day_of_Month,
    count(*) AS Total_Customers
FROM V_SubsPlan
WHERE
    plan_name = 'trial'
GROUP BY
    First_Day_of_Month
```

The table shows the monthly distribution of customers with trial plan. Overall, there are **around 80 trial plan customers** each month. The **highest** number of customers is on **March** and the **lowest** is on **February**.



	First_Day_of_Month	Total_Customers
1	2020-01-01	88
2	2020-02-01	68
3	2020-03-01	94
4	2020-04-01	81
5	2020-05-01	88
6	2020-06-01	79
7	2020-07-01	89
8	2020-08-01	88
9	2020-09-01	87
10	2020-10-01	79
11	2020-11-01	75
12	2020-12-01	84



Data Analysis

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
    vsp.plan_name AS Plan_Name,
    count(*) AS [Counts of Each Plan Name After Year 2020]
FROM
    V_SubsPlan AS vsp
WHERE
    strftime('%Y',start_date) > '2020'
GROUP BY
    vsp.plan_name
```

	Plan_Name	Counts of Each Plan Name After Year 2020
1	basic monthly	8
2	churn	71
3	pro annual	63
4	pro monthly	60

The table shows the total customers based on plan name after the year 2020. **After the year 2020, most customers decided to churn or cancel the Foodie-Fi service**





Data Analysis

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

```
SELECT
    tcwhc AS [Total Customers Who Have Churned],
    tc AS [Total Customers],
    round(CAST(tcwhc AS FLOAT)/tc*100,1) AS [Percentage of Customers Who Have Churned]
FROM (
    SELECT
        count(*) AS tcwhc,
        (SELECT
            count(DISTINCT s.customer_id) AS [Total Customers]
            FROM subscriptions AS s) AS tc
    FROM
        V_SubsPlan AS vsp
    WHERE
        vsp.plan_name = 'churn'
) AS c
```

Out of 1.000 total customers, **30,7%** had churned.

Total Customers Who Have Churned	Total Customers	Percentage of Customers Who Have Churned
307	1000	30.7



Data Analysis



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

```
WITH end_trial AS (
  SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.start_date,
    DATE(vsp.start_date, '+7 days') AS [free_trial_end]
  FROM
    V_SubsPlan AS vsp
  WHERE
    vsp.plan_id = 0
),
straight_churn AS (
  SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.start_date,
    et.free_trial_end,
    CASE
      WHEN DATE(vsp.start_date) = DATE(et.free_trial_end) AND vsp.plan_id = 4 THEN 'YES'
        ELSE 'NO'
    END AS [straight_churn]
  FROM
    V_SubsPlan AS vsp
  INNER JOIN
    end_trial AS et
  ON
    vsp.customer_id = et.customer_id
  WHERE
    straight_churn = 'YES'
)
```

```
SELECT
  tsc.total_straight_churn AS [Total Customers Straight Churn],
  tsc.total_customers AS [Total Customers],
  round(CAST(tsc.total_straight_churn AS FLOAT)/tsc.total_customers*100,1) AS [Percentage of Customers Who Have Straight Churned]
FROM (
  SELECT
    count(*) AS [total_straight_churn],
    (SELECT
      count(DISTINCT s.customer_id) AS [Total Customers]
      FROM subscriptions AS s) AS [total_customers]
    FROM
      straight_churn AS sc
  ) AS tsc
```

Total Customers Straight Churn	Total Customers	Percentage of Customers Who Have Straight Churned
92	1000	9.2

Out of 1.000 total customers, **9,2%** had churned straight after the initial free trial.





Data Analysis

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

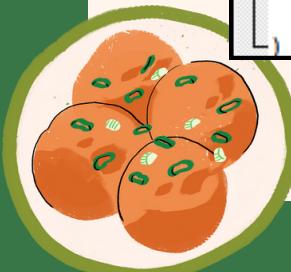
```
WITH end_trial AS (
  SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.plan_name,
    vsp.start_date,
    DATE(vsp.start_date, '+7 days') AS [free_trial_end]
  FROM
    V_SubsPlan AS vsp
  WHERE
    vsp.plan_id = 0
),
```

plan_name	total_customers_join_plans	total_customers	Percentage of Customers who join plans
basic monthly	546	1000	54.6
pro annual	37	1000	3.7
pro monthly	325	1000	32.5

After initial free trial, **54.6% customers continued to basic monthly plan** and **3.7% customers upgraded to pro annual plan.**

```
join_plans AS (
  SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.plan_name,
    vsp.start_date,
    et.free_trial_end,
    CASE
      WHEN DATE(vsp.start_date) = DATE(et.free_trial_end) AND vsp.plan_id <> 4 THEN 'YES'
      ELSE 'NO'
    END AS [join_plans]
  FROM
    V_SubsPlan AS vsp
  INNER JOIN
    end_trial AS et
  ON
    vsp.customer_id = et.customer_id
  WHERE
    join_plans = 'YES'
)
```

```
SELECT
  tjp.plan_name,
  tjp.total_customers_join_plans,
  tjp.total_customers,
  round(CAST(tjp.total_customers_join_plans AS FLOAT)/tjp.total_customers*100,1) AS [Percentage of Customers who join plans]
FROM (
  SELECT
    jp.plan_name,
    count(*) AS [total_customers_join_plans],
    (SELECT
      count(DISTINCT s.customer_id) AS [Total Customers]
      FROM subscriptions AS s) AS [total_customers]
  FROM
    join_plans AS jp
  GROUP BY
    jp.plan_name
) AS tjp
```





Data Analysis

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

```
WITH maxdate AS (SELECT
    vsp.customer_id,
    max(vsp.start_date) AS maxsd
  FROM
    V_SubsPlan AS vsp
  WHERE
    DATE(vsp.start_date) <= '2020-12-31'
  GROUP BY
    vsp.customer_id
)
SELECT
    pn.plan_name AS Plan_Name,
    pn.customer_count AS Total_Customers,
    round(CAST(pn.customer_count AS FLOAT)/pn.total_customers*100,1) AS [Percentage of Customers]
FROM (
  SELECT
    vsp.plan_name,
    count(*) AS [customer_count],
    (SELECT
      count(DISTINCT s.customer_id) AS [Total Customers]
      FROM subscriptions AS s) AS [total_customers]
  FROM
    V_SubsPlan AS vsp
  INNER JOIN
    maxdate AS md
  ON
    vsp.customer_id = md.customer_id
  WHERE
    DATE(vsp.start_date) = DATE(md.maxsd)
  GROUP BY
    vsp.plan_name
) AS pn
```

```
    ) AS pn
```

	Plan_Name	Total_Customers	Percentage of Customers
1	basic monthly	224	22.4
2	churn	236	23.6
3	pro annual	195	19.5
4	pro monthly	326	32.6
5	trial	19	1.9

At the **2020-12-31**, out of 1.000 customers, **32,6%** (the **majority**) were using **pro monthly plan**, **23,6%** decided to **churn**, and **1,9%** were in the initial free trial phase.





Data Analysis

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

```
SELECT  
    count(DISTINCT vsp.customer_id) AS [customers upgraded to annual plan]  
FROM  
    V_SubsPlan AS vsp  
WHERE  
    strftime('%Y',vsp.start_date) = '2020' AND vsp.plan_name = 'pro annual'
```

customers upgraded to annual plan

195

There are **195** customers who upgraded to **pro annual plan** in 2020.



Data Analysis



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

```
WITH joindate AS (SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.plan_name,
    vsp.start_date AS [join_date]
  FROM
    V_SubsPlan AS vsp
 WHERE
    vsp.plan_id = 0
)
SELECT
    CAST(avg(dap.days_to_annual_plan) AS INTEGER) AS [average days to upgrade to annual plan]
FROM (
  SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.plan_name,
    vsp.start_date,
    jd.join_date,
    julianday(vsp.start_date) - julianday(jd.join_date) AS [days_to_annual_plan]
  FROM
    V_SubsPlan AS vsp
  INNER JOIN
    joindate AS jd
  ON
    vsp.customer_id = jd.customer_id
  WHERE
    vsp.plan_name = 'pro annual'
) AS dap
```

average days to upgrade to annual plan
104

The **average** days a customer take to **annual plan** is **104 days** from the day the join Foodie-Fi.





Data Analysis

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

```

WITH joindate AS (SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.plan_name,
    vsp.start_date AS [join_date]
FROM
    V_SubsPlan AS vsp
WHERE
    vsp.plan_id = 0
)
SELECT
    CASE
        WHEN ap.days_to_annual_plan BETWEEN 0 AND 30 THEN '0-30 days'
        WHEN ap.days_to_annual_plan BETWEEN 31 AND 60 THEN '31-60 days'
        WHEN ap.days_to_annual_plan BETWEEN 61 AND 90 THEN '61-90 days'
        WHEN ap.days_to_annual_plan BETWEEN 91 AND 120 THEN '91-120 days'
        WHEN ap.days_to_annual_plan BETWEEN 121 AND 150 THEN '121-150 days'
        WHEN ap.days_to_annual_plan BETWEEN 151 AND 180 THEN '151-180 days'
        ELSE '180+ days'
    END AS [periods_to_annual_plan],
    count(*) AS [total customers to annual plan based on periods]
FROM (

```



```

SELECT
    vsp.customer_id,
    vsp.plan_id,
    vsp.plan_name,
    vsp.start_date,
    jd.join_date,
    julianday(vsp.start_date) - julianday(jd.join_date) AS [days_to_annual_plan]
FROM
    V_SubsPlan AS vsp
INNER JOIN
    joindate AS jd
ON
    vsp.customer_id = jd.customer_id
WHERE
    vsp.plan_name = 'pro annual'
) AS ap
GROUP BY
    periods_to_annual_plan
ORDER BY
    MIN(days to annual plan)

```

	periods_to_annual_plan	total customers to annual plan based on periods
1	0-30 days	49
2	31-60 days	24
3	61-90 days	34
4	91-120 days	35
5	121-150 days	42
6	151-180 days	36
7	180+ days	38

Overall, the **distribution of total customers** who upgrade to **annual plan** is fairly even across each 30-day period.



Data Analysis



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

```
WITH promonth AS (SELECT
    vsp.customer_id,
    vsp.plan_name,
    vsp.start_date AS [start_date_promonth]
FROM
    V_SubsPlan AS vsp
WHERE
    strftime('%Y',vsp.start_date) = '2020' AND vsp.plan_name = 'pro monthly'
),
basmonth AS (SELECT
    vsp.customer_id,
    vsp.plan_name,
    vsp.start_date AS [start_date_basmonth]
FROM
    V_SubsPlan AS vsp
WHERE
    strftime('%Y',vsp.start_date) = '2020' AND vsp.plan_name = 'basic monthly'
)
```

customers who downgrade from pro month to basic month in 2020

0

There are **no customer who downgraded** from pro monthly plan to basic monthly plan in 2020.

```
SELECT
    count(*) AS [customers who downgrade from pro month to basic month in 2020]
FROM
    promonth AS pm
INNER JOIN
    basmonth AS bm
ON
    pm.customer_id = bm.customer_id
WHERE
    DATE(pm.start_date_promonth) < DATE(bm.start_date_basmonth)
```



★ THANK YOU ★

