SQL Case Study - Foodie-Fi

By Naveed Khan

FOODIE-FI

- Launched in 2020 by Danny and his team, Foodie-Fi is a subscription-based streaming platform dedicated to all things food.
- Filling a market gap, it offers a Netflix-like experience focused solely on cooking shows and culinary adventures.
- By embracing a data-driven approach, Foodie-Fi ensures its content and services cater directly to your taste buds

CASE STUDY GOALS

CUSTOMER INSIGHTS: Analyze total users, location, churn rates, and subscription preferences (monthly vs. annual) to understand who your customers are and how they subscribe.

SUBSCRIPTION PATTERN ANALYSIS: Investigate trial sign-ups, plan start dates, and upgrade/downgrade trends to identify subscription behavior patterns.

RETENTION & CONVERSION: Evaluate churn rates, trial-to-paid conversion rates, and analyze post-trial plan preferences to improve user retention and convert more trial users.

DATA REPRESENTATION:FOODIE-FI SUBSCRIPTION DATABASE

PLANS TABLE: This table details the various subscription options offered by Foodie-Fi. It includes:

PLAN ID: Unique identifier for each plan

(e.g., trial, basic monthly, pro monthly, pro annual)

NAME: Descriptive name of the subscription plan

(e.g., "Basic Monthly")

PRICE: Monthly or annual cost associated with the plan

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

CONT...

SUBSCRIPTIONS TABLE: This table captures individual user subscriptions with the following data points:

CUSTOMER_ID: Unique identifier for each customer

PLAN_ID: Connects a customer to their specific subscription plan (references the Plans table)

START_DATE: Date the customer's subscription began

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

BUSINESS CHALLENGE

SUBSCRIBER ACQUISITION AND RETENTION:

- Attract and maintain subscribers in a fiercely competitive market.
- Gain insights into customer preferences and refine content selection.
- Improve user experience to decrease customer turnover rates.

DATA-DRIVEN DECISION-MAKING:

- Leverage data analytics for informed decision-making.
- Develop resilient data infrastructure and integrate predictive analytics.
- Cultivate a culture of data-driven innovation for sustainable growth.

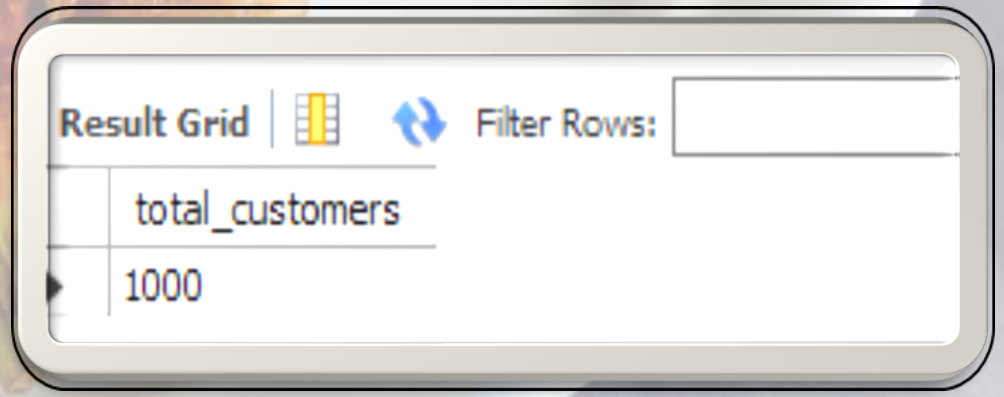
CASE STUDY QUESTIONS

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

SQL CODE:

```
SELECT
        COUNT(DISTINCT customer_id) AS total_customers
FROM subscriptions;
```

RESULT:



FINDINGS:

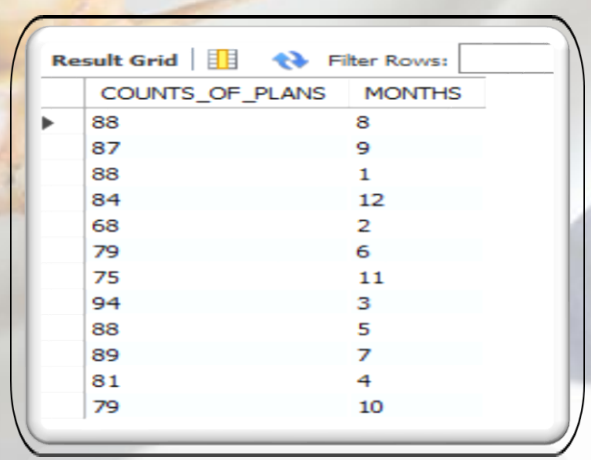
- ❑ **Foodie-Fi has a customer base of 1,000 users.**
- ❑ **This is a good starting point to understand the overall reach of the service.**

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?

SQL CODE:

```
SELECT
	COUNT(PLAN_ID) AS COUNTS_OF_PLANS,
	MONTH(START_DATE) AS MONTHS
FROM SUBSCRIPTIONS
	GROUP BY MONTHS, PLAN_ID
HAVING PLAN_ID=0;
```

RESULT:


Result Grid | Filter Rows:

COUNTS_OF_PLANS	MONTHS
88	8
87	9
88	1
84	12
68	2
79	6
75	11
94	3
88	5
89	7
81	4
79	10

FINDINGS:

Customer engagement levels exhibit distinct peaks in March and August, according to the monthly distribution of trial plan start dates.

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.

SQL CODE:

```
SELECT
S.PLAN_ID,
P.PLAN_NAME,
COUNT(P.PLAN_NAME) AS EVENT_COUNT
FROM SUBSCRIPTIONS S INNER JOIN PLANS P
ON S.PLAN_ID = P.PLAN_ID
WHERE YEAR(S.START_DATE) > 2020
GROUP BY S.PLAN_ID, P.PLAN_NAME;
```

RESULT:

Result Grid | Filter Rows:

PLAN_ID	PLAN_NAME	EVENT_COUNT
4	churn	71
2	pro monthly	60
3	pro annual	63
1	basic monthly	8

FINDINGS:

Churn events post-2020 highlight a notable increase in subscription cancellations, possibly indicating shifting consumer preferences or dissatisfaction with services.

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

SQL CODE:

```
SELECT
COUNT(DISTINCT customer_id) as customer_count, ROUND((count(DISTINCT
customer_id)/(SELECTCOUNT(DISTINCT customer_id)
FROM subscriptions)) * 100, 1)
AS percentage
FROM subscriptions
WHERE plan_id = 4;
```

RESULT:

Result Grid | Filter Rows:

	customer_count	percentage
▶	307	30.7

FINDINGS: Foodie-Fi's churn rate stands at approximately 30.7%, with 307 customers having ended their subscription, warranting immediate attention to retention strategies.

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

SQL CODE:

```
WITH cte_churn AS (
    SELECT *, LAG(plan_id, 1) OVER(PARTITION BY
        customer_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 perc_churn
FROM cte_churn
WHERE plan_id = 4
AND prev_plan = 0;
```

RESULT:

Result Grid | Filter Rows:

	cnt_churn	perc_churn
▶	92	9

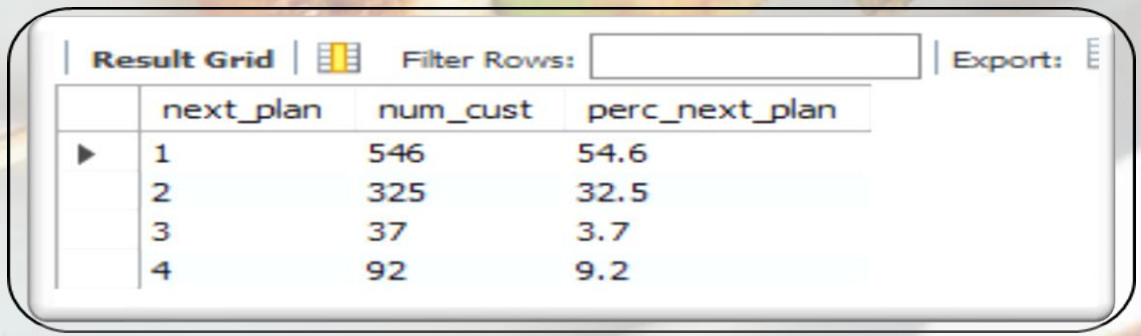
FINDINGS:Post-2020, churn events escalate, revealing a significant surge in subscription cancellations, with Pro monthly and Pro annual plans maintaining traction, while Basic monthly plan activations dwindle.

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

SQL CODE:

```
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;
```

RESULT:



Result Grid | Filter Rows: | Export:

next_plan	num_cust	perc_next_plan
1	546	54.6
2	325	32.5
3	37	3.7
4	92	9.2

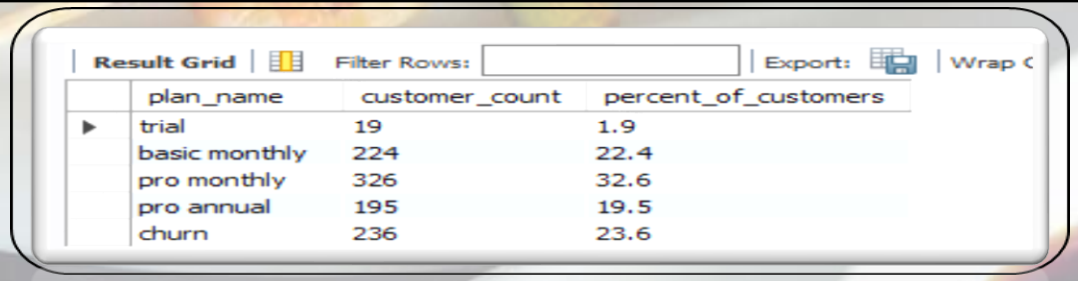
FINDINGS: After the free trial, the Basic monthly plan is favored by 54.6% of customers, while 32.5% opt for the Pro monthly plan. A mere 3.7% choose the Pro annual plan, with 9.2% churning from the service.

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

SQL CODE:

```
WITH My_CTE AS (
SELECT *,
     ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY start_date DESC) as rwnmbr
FROM subscriptions
WHERE start_date <= '2020-12-31'
)
SELECT
    plan_name,
    COUNT(customer_id) as customer_count,
     ROUND((COUNT(customer_id) / (SELECT COUNT(DISTINCT customer_id) FROM My_CTE)) * 100, 1) as
     percent_of_customersFROM My_CTE mc
INNER JOIN plans as P ON mc.plan_id = P.plan_id
WHERE rwnmbr = 1
GROUP BY plan_name;
```

RESULT:



Result Grid | Filter Rows: | Export: | Wrap

plan_name	customer_count	percent_of_customers
trial	19	1.9
basic monthly	224	22.4
pro monthly	326	32.6
pro annual	195	19.5
churn	236	23.6

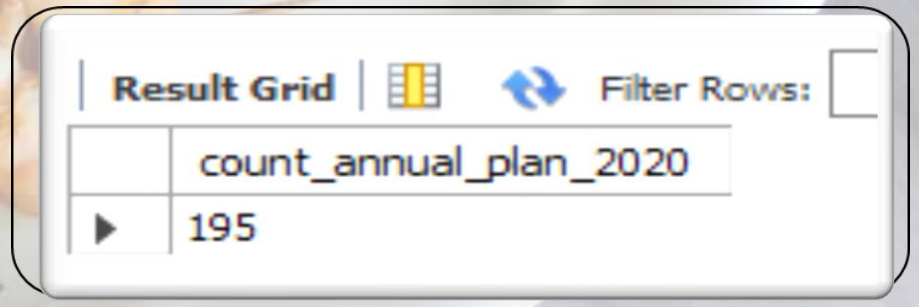
FINDINGS: As of December 31, 2020, Pro monthly holds the highest subscription rate at 32.6%, followed by Basic monthly at 22.4%. Pro annual and churn plans each represent 19.5% and 23.6% of the customer base, respectively, while the trial plan has the lowest subscription rate at 1.9%.

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

SQL CODE:

```
SELECT
  count(*) AS count_annual_plan_2020
FROM subscriptions
WHERE Year(start_date) = 2020 and plan_id = 3;
```

RESULT:



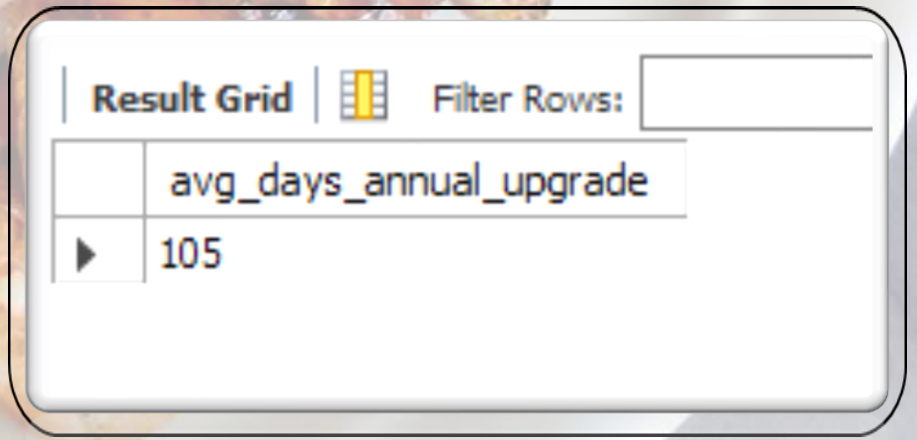
FINDINGS: In 2020, 195 customers upgraded to an annual plan, showcasing a notable preference for long-term subscription commitments, likely driven by perceived value and cost-efficiency.

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

SQL CODE:

```
WITH trail_plan AS (
 SELECT
customer_id,
start_date AS trail_dates
FROM subscriptions
WHERE plan_id=0),
annual_plan as (
select
customer_id,
Start_date as annual_dates
  from subscriptions
  where plan_id=3)
SELECT  ROUND(AVG(DATEDIFF(annual_dates, trail_dates)),0) AS
avg_days_annual_upgrade
FROM annual_plan ap JOIN trail_plan tp
ON ap.customer_id = tp.customer_id;
```

RESULT:



FINDINGS: The average time for customers to upgrade from the trial plan to an annual subscription on Foodie-Fi is approximately105 days. This indicates a considerable consideration period before committing to a long-term subscription, highlighting the importance of strategic engagement and marketing efforts to prompt timely upgrades.

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

SQL CODE:

```
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 binsFROM day_period)
    SELECT CONCAT((bins * 30) + 1, ' - ', (bins + 1) * 30, ' days ') AS days,
   COUNT(diff) AS total
   FROM bins GROUP BY bins;
```

RESULT:

Result Grid | Filter Rows:

avg_days_annual_upgrade
105

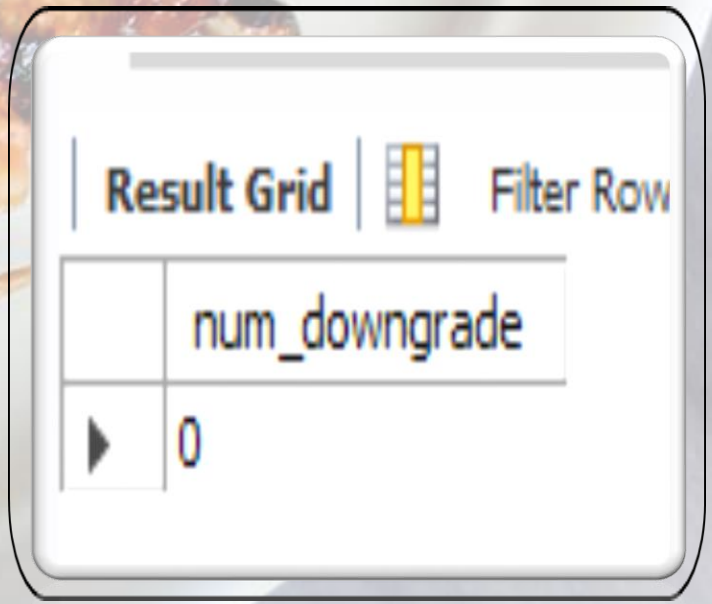
FINDINGS: Effective engagement strategies within the first 30 days are pivotal for encouraging customers to upgrade from the trial plan to an annual subscription, while sustained efforts between 31-180 days can maintain consistent engagement levels before upgrades diminish beyond 180 days.

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

SQL CODE:

```
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';
```

RESULT:



FINDINGS: In 2020, the absence of downgrades from the Pro monthly to the Basic monthly plan highlights the effectiveness of delivering compelling features to maintain satisfaction and perceived value among subscribers.