SQL Portfolio Project

Submitted By: Fariha Arif

Introduction:

In this analysis, we're looking into Foodie-Fi's subscription data. Foodie-Fi is a subscription service for food lovers, offering different plans for its users. By using SQL queries, we'll dig into the data to understand how customers are using the service and what their preferences are when it comes to subscription plans.

Q#01

How many customers has Foodie-Fi ever had?

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

Result Grid | Filter Rows:

total_customers
1000

Q#02

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 DATE_FORMAT(start_date, '%Y-%m-01') AS start_of_month,
    COUNT(*) AS trial_plan_subscriptions
FROM subscriptions
WHERE plan_id = 0
GROUP BY start_of_month
ORDER BY start_of_month;
```

Q#03

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_name,
    COUNT(*) AS event_count
FROM subscriptions AS s
JOIN plans AS p ON s.plan_id = p.plan_id
WHERE s.start_date > '2020-01-01'
GROUP BY p.plan_name;
```

Result Grid | Filter Rows:

plan_name	event_count
trial	997
basic monthly	546
pro annual	258
churn	307
pro monthly	539

Q#04

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

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

Result Grid | Filter Rows: | Export:

churned_customer_count	churned_percentage
307	30.7

Q#05

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

```
SELECT
   COUNT(DISTINCT s1.customer_id) AS churned_after_trial_count,
   ROUND((COUNT(DISTINCT s1.customer_id) / (SELECT COUNT(DISTINCT
customer_id)
FROM subscriptions WHERE plan_id = 0)) * 100) AS churned_after_trial_percentage
FROM
   subscriptions AS s1
JOIN
   subscriptions AS s2 ON s1.customer_id = s2.customer_id AND s2.plan_id = 4
WHERE
   s1.plan_id = 0 AND
   s1.start_date = (SELECT MIN(start_date) FROM subscriptions WHERE customer_id =
s1.customer_id);
```

Result Grid | Filter Rows: | Export: | Wrap

churned_after_trial_count	churned_after_trial_percentage
307	31

Q#06

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

```
SELECT
   COUNT(*) AS post_trial_plan_count,
   ROUND((COUNT(*) / (SELECT COUNT(*) FROM subscriptions WHERE plan_id != 0)) *
100) AS post_trial_plan_percentage
FROM
   subscriptions
WHERE
   plan_id != 0;
```

Result Grid | Filter Rows: | Export:

post_trial_plan_count	post_trial_plan_percentage
1650	100

Q#07

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

```
SELECT
   p.plan_name,
   COUNT(DISTINCT s.customer_id) AS customer_count,
   ROUND((COUNT(DISTINCT s.customer_id) / (SELECT COUNT(DISTINCT customer_id)
FROM subscriptions WHERE start_date <= '2020-12-31')) * 100) AS percentage
FROM
   subscriptions AS s
JOIN
   plans AS p ON s.plan_id = p.plan_id
WHERE
   s.start_date <= '2020-12-31'
GROUP BY
   p.plan_name;
```

Result Grid | Filter Rows: | Export:

plan_name	customer_count	percentage
basic monthly	538	54
churn	236	24
pro annual	195	20
pro monthly	479	48
trial	1000	100

Q#08

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

```
SELECT COUNT(DISTINCT customer_id) AS upgraded_to_annual
FROM subscriptions
WHERE plan_id = 3
AND YEAR(start_date) = 2020;
```

Result Grid | Filter Rows:

upgraded_to_annual
195

Q#09

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

```
SELECT AVG(DATEDIFF(
   (SELECT MIN(start_date) FROM subscriptions s2 WHERE s1.customer_id = s2.customer_id
AND s2.plan_id = 3),
   (SELECT MIN(start_date) FROM subscriptions s3 WHERE s1.customer_id =
s3.customer_id)
)) AS average_days_to_upgrade
FROM subscriptions s1
WHERE s1.plan_id != 3;
```

Result Grid | Filter Rows:

average_days_to_upgrade
116.6115

Q#10

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

```
SELECT
   CONCAT((FLOOR(difference / 30) * 30) + 1, '-', (FLOOR(difference / 30) * 30) + 30) AS
period,
   AVG(difference) AS average_days_to_upgrade
FROM (
   SELECT
      DATEDIFF(
         (SELECT MIN(start_date) FROM subscriptions s2 WHERE s1.customer_id =
s2.customer_id AND s2.plan_id = 3),
         (SELECT MIN(start_date) FROM subscriptions s3 WHERE s1.customer_id =
s3.customer_id)
      ) AS difference
```

```
    FROM
        subscriptions s1
    WHERE
        s1.plan_id != 3
    GROUP BY
        s1.customer_id
) AS subquery
GROUP BY
    FLOOR(difference / 30), difference;
```

Q#11

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

```
SELECT
    COUNT(DISTINCT customer_id) AS downgrade_count
FROM
    subscriptions
WHERE
    plan_id = 1
    AND customer_id IN (
        SELECT DISTINCT customer_id
        FROM subscriptions
        WHERE plan_id = 2
         AND YEAR(start_date) = 2020
    );
```

Result Grid | Filter F

	downgrade_count
▶	163

Conclusion:

By using SQL to analyze Foodie-Fi's subscription data, we've gained valuable insights into customer behavior and preferences. This information can help Foodie-Fi make informed decisions about their plans and offerings to better serve their customers.