Case Study #3 - Foodie-Fi

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

```
SELECT
    COUNT(DISTINCT customer_id) AS customer_count
FROM subscriptions
```

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
    MONTH(start_date) AS month_of_year,
    COUNT(*) AS trial_plan_count
FROM subscriptions AS s
JOIN plans AS p ON s.plan_id = p.plan_id
WHERE plan_name = 'trial'
GROUP BY month_of_year
ORDER BY month_of_year
```

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
    s.plan_id,
    plan_name,
    COUNT(*) AS plan_count
FROM subscriptions AS s
JOIN plans AS p ON s.plan_id = p.plan_id
WHERE YEAR(start_date) > 2020
GROUP BY s.plan_id, plan_name
ORDER BY s.plan_id
```

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

```
SELECT
    SUM(
        IF(plan_name = 'churn',1,0)
        ) AS churn_count,
    ROUND(SUM(
        IF(plan_name = 'churn',1,0)
        )/COUNT(DISTINCT customer_id) *100,1)  AS churn_count
FROM subscriptions AS s
JOIN plans AS p ON s.plan_id = p.plan_id
```

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

```
WITH cte_rank AS (
    SELECT
        *,
        ROW_NUMBER() OVER (PARTITION BY customer_id) as ranking
    FROM subscriptions
)



SELECT
    COUNT(*) as customer_count,
    ROUND(COUNT(*)/(SELECT COUNT(DISTINCT customer_id) from subs
FROM cte_rank
WHERE customer_id IN
        (SELECT
            customer_id
        FROM cte_rank
        WHERE (ranking = 1 and plan_id = 0))
    AND ranking = 2 and plan_id = 4
```

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

```
SELECT
    plan_id,
    COUNT(customer_id) as customer_count,
    ROUND(COUNT(customer_id)/(SELECT COUNT(DISTINCT customer_id)
FROM
    (SELECT
        *,
        LAG(plan_id) OVER (PARTITION BY customer_id) as previous
    FROM subscriptions) as sub
WHERE previous_plan = 0
GROUP BY plan_id
ORDER BY plan_id
```

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

```
SELECT
    plan_id,
    COUNT(customer_id) AS customer_count,
    ROUND(COUNT(customer_id)/(SELECT COUNT(DISTINCT customer_id)
FROM (
    SELECT
        *,
        LAST_VALUE(plan_id) OVER(PARTITION BY customer_id) as la
    FROM subscriptions
    WHERE start_date <= '2020-12-31') AS subquery
WHERE plan_id = last_plan_id
GROUP BY plan_id
ORDER BY plan_id
```

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

```
SELECT
    COUNT(DISTINCT customer_id) as customer_count
FROM subscriptions
WHERE YEAR(start_date) = 2020 and plan_id = 3
```

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

```
SELECT
    ROUND(AVG(TIMESTAMPDIFF(DAY, start_date, annual_date)),0) as
FROM subscriptions AS s1
JOIN (
    SELECT
        customer_id,
        start_date AS annual_date
    FROM subscriptions
    WHERE plan_id = 3) AS s2 ON s1.customer_id = s2.customer_id
WHERE plan_id = 0
```

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

```
SELECT
    CONCAT(TIMESTAMPDIFF(MONTH, start_date, annual_date) *30,' -
    TIMESTAMPDIFF(MONTH, start_date, annual_date) as period_mont
    COUNT(*) as customer_count
FROM subscriptions AS s1
JOIN (
    SELECT
        customer_id,
        start_date AS annual_date
    FROM subscriptions
    WHERE plan_id = 3) AS s2 ON s1.customer_id = s2.customer_id
WHERE plan_id = 0
```

```
GROUP BY period,period_month
ORDER BY period_month
```

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

```
SELECT
    COUNT(DISTINCT customer_id) as customer_count
FROM
    (SELECT
        *,
        LAG(plan_id) OVER (PARTITION BY customer_id) as previous
    FROM subscriptions) as sub
WHERE plan_id = 2 and previous_plan = 3
```