

1. Problem Statement – 1

How many customers has Foodie-Fi ever had?

Solution - 1

```
SELECT COUNT(DISTINCT customer_id) AS unique_customer  
FROM dbo.subscriptions;
```

2. Problem Statement - 2

What is the monthly distribution of trial plan start date values for our dataset?

Solution - 2

```
SELECT DATE_PART('month',start_date) AS month_date,  
       TO_CHAR(start_date, 'Month') AS month_name,  
       COUNT(*) AS trial_subscriptions  
FROM dbo.subscriptions s  
JOIN dbo.plans p  
ON s.plan_id = p.plan_id  
WHERE s.plan_id = 0  
GROUP BY DATE_PART('month',start_date),  
         TO_CHAR(start_date, 'Month')  
ORDER BY month_date ASC;
```

3. Problem Statement - 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.

Solution - 3

```
SELECT p.plan_id,  
       p.plan_name,  
       COUNT(*) AS events  
FROM dbo.subscriptions s  
JOIN dbo.plans p  
ON s.plan_id = p.plan_id  
WHERE s.start_date >= '2021-01-01'  
GROUP BY p.plan_id, p.plan_name  
ORDER BY p.plan_id;
```

4. Problem Statement - 4

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

Solution - 4

```
SELECT COUNT(*) AS churn_count,  
       ROUND(100 * COUNT(*)::NUMERIC / (SELECT COUNT(DISTINCT  
       customer_id) FROM    dbo.subscriptions),1) AS churn_percentage  
FROM dbo.subscriptions s  
JOIN dbo.plans p  
ON s.plan_id = p.plan_id  
WHERE s.plan_id = 4;
```

5. Problem Statement - 5

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

Solution - 5

```
WITH ranking AS (  
  SELECT s.customer_id,  
         s.plan_id,  
         p.plan_name,  
         ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY  
                               s.plan_id) AS plan_rank  
  FROM dbo.subscriptions s  
  JOIN dbo.plans p  
  ON s.plan_id = p.plan_id)  
  
  SELECT COUNT(*) AS churn_count,  
         ROUND(100 * COUNT(*) / (SELECT COUNT(DISTINCT customer_id) FROM  
                                   dbo.subscriptions),0) AS churn_percentage  
  FROM ranking  
  WHERE plan_id = 4 -- Filter to churn plan  
  AND plan_rank = 2
```

6. Problem Statement - 6

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

Solution - 6

```
WITH next_plan_cte AS (  
    SELECT customer_id,  
           plan_id,  
           LEAD(plan_id, 1) OVER( PARTITION BY customer_id ORDER BY plan_id) as  
           next_plan  
    FROM dbo.subscriptions)  
  
SELECT next_plan,  
       COUNT(*) AS conversions,  
       ROUND(100 * COUNT(*)/ (SELECT COUNT(DISTINCT customer_id) FROM  
       dbo.subscriptions),1) AS conversion_percentage  
FROM next_plan_cte  
WHERE next_plan IS NOT NULL  
AND plan_id = 0  
GROUP BY next_plan  
ORDER BY next_plan;
```

7. Problem Statement - 7

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

Solution - 7

```
WITH next_plan AS(
SELECT customer_id,
       plan_id,
       start_date,
       LEAD(start_date, 1) OVER(PARTITION BY customer_id ORDER BY
       start_date) as next_date FROM dbo.subscriptions
WHERE start_date <= '2020-12-31'),

customer_breakdown AS (
SELECT plan_id,
       COUNT(DISTINCT customer_id) AS customers
FROM next_plan
WHERE (next_date IS NOT NULL AND (start_date < '2020-12-31'
       AND next_date > '2020-12-31'))
       OR (next_date IS NULL AND start_date < '2020-12-31')
GROUP BY plan_id)
SELECT plan_id,
       customers,
       ROUND(100 * customers / (SELECT COUNT(DISTINCT customer_id)
FROM dbo.subscriptions),1) AS percentage
FROM customer_breakdown
GROUP BY plan_id, customers
ORDER BY plan_id;

SELECT plan_id,
       customers,
       ROUND(100 * customers / (SELECT COUNT(DISTINCT customer_id) FROM
dbo.subscriptions),1) AS percentage
FROM customer_breakdown
GROUP BY plan_id, customers
ORDER BY plan_id;
```

8. Problem Statement - 8

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

Solution - 8

```
SELECT COUNT(DISTINCT customer_id) AS unique_customer
FROM dbo.subscriptions
WHERE plan_id = 3
AND start_date <= '2020-12-31';
```

9. Problem Statement - 9

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

Solution - 9

```
-- Filter results to customers at trial plan = 0
WITH trial_plan AS (
SELECT customer_id,
       start_date AS trial_date
FROM dbo.subscriptions
WHERE plan_id = 0),
-- Filter results to customers at pro annual plan = 3
annual_plan AS
(SELECT customer_id,
       start_date AS annual_date
FROM dbo.subscriptions
WHERE plan_id = 3)

SELECT ROUND(AVG(annual_date - trial_date),0) AS avg_days_to_upgrade
FROM trial_plan tp
JOIN annual_plan ap
ON tp.customer_id = ap.customer_id;
```

10. Problem Statement - 10

Can you further breakdown this average value into 30-day periods?

Solution - 10

```
-- Filter results to customers at trial plan = 0
WITH trial_plan AS (
  SELECT customer_id,
         start_date AS trial_date
  FROM dbo.subscriptions
  WHERE plan_id = 0),
-- Filter results to customers at pro annual plan = 3
annual_plan AS (
  SELECT customer_id,
         start_date AS annual_date
  FROM dbo.subscriptions
  WHERE plan_id = 3),
-- Sort values above in buckets of 12 with range of 30 days each
bins AS (
  SELECT WIDTH_BUCKET(ap.annual_date - tp.trial_date, 0, 360, 12) AS
         avg_days_to_upgrade
  FROM trial_plan tp
  JOIN annual_plan ap
  ON tp.customer_id = ap.customer_id)

SELECT ((avg_days_to_upgrade - 1) * 30 || ' - ' || (avg_days_to_upgrade) * 30)
|| ' days' AS breakdown,
       COUNT(*) AS customers
FROM bins
GROUP BY avg_days_to_upgrade
ORDER BY avg_days_to_upgrade
```

11. Problem Statement - 11

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

Solution - 11

```
WITH next_plan_cte AS (  
    SELECT customer_id,  
           plan_id,  
           start_date,  
           LEAD(plan_id, 1) OVER( PARTITION BY customer_id ORDER BY plan_id)  
           as next_plan  
FROM dbo.subscriptions)  
SELECT COUNT(*) AS downgraded  
FROM next_plan_cte  
WHERE start_date <= '2020-12-31'  
AND plan_id = 2  
AND next_plan = 1;
```