**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 subscriptions)\*100) as percentage

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 subscriptions)\*100,1) as percentage

FROM

(SELECT

\*,

LAG(plan\_id) OVER (PARTITION BY customer\_id) as previous\_plan

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 subscriptions)\*100,1) as percentage

FROM (

SELECT

\*,

LAST\_VALUE(plan\_id) OVER(PARTITION BY customer\_id) as last\_plan\_id

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 avg\_date\_join

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) + 1)\*30,' days') as period,

TIMESTAMPDIFF(MONTH, start\_date, annual\_date) as period\_month,

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\_plan

FROM subscriptions) as sub

WHERE plan\_id = 2 and previous\_plan = 3