



INTRODUCTION:

Foodie-Fi is a website where one can sign up to watch many cooking related shows from all over the world. Danny, who launched Foodie-Fi with few smart friends and started selling monthly and annual subscriptions, giving their customers unlimited on-demand access to exclusive food videos.

Danny created Foodie-Fi with a data driven mindset and wanted to ensure all future investment decisions and new features were decided according to what people desire to watch and what they like.

Therefore needed to analyze the data.

Data Analysis:

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

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

Result Grid	Filter Rows:	Export:
total_customer		
1000		

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 start_month,COUNT(*) AS num_of_customer
FROM subscriptions s
JOIN plans p ON s.plan_id=p.plan_id
WHERE p.plan_name='trial'
GROUP BY start_month
ORDER BY start_month;
```

Result Grid	Filter Rows:	Export:
start_month	num_of_customer	
1	88	
2	68	
3	94	
4	81	
5	88	
6	79	
7	89	
8	88	
9	87	
10	79	
11	75	
12	84	

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

```

Result Grid		
	plan_name	Number_of_Events
▶	churn	71
	pro monthly	60
	pro annual	63
	basic monthly	8

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

```

SELECT COUNT(customer_id) AS Churned_Customers_Count,
ROUND(COUNT(customer_id) / (SELECT COUNT(DISTINCT customer_id) FROM subscriptions) * 100, 1) AS Churn_Custom_Percent
FROM subscriptions
WHERE plan_id=(SELECT plan_id FROM plans WHERE plan_name='churn');

```

Result Grid		
	Churned_Customers_Count	Churned_Customers_Percentage
▶	307	30.7

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

```

WITH churn_cte AS(
SELECT * ,
LAG(plan_id,1) OVER(PARTITION BY customer_id ) AS previous_plan
FROM subscriptions)
SELECT COUNT(previous_plan) AS number_of_churn,
ROUND(COUNT(*)/(SELECT COUNT(DISTINCT customer_id) FROM subscriptions)*100,0) AS churn_percentage
FROM churn_cte
WHERE plan_id=(SELECT plan_id FROM plans WHERE plan_name='churn') AND previous_plan=0;

```

Result Grid			Filter Rows:	Export:
	number_of_churn	churn_percentage		
▶	92	9		

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

```
WITH cte_for_nextplan AS(
SELECT *,LEAD(plan_id,1) OVER(PARTITION BY customer_id) AS nextplan
FROM subscriptions)
SELECT nextplan,count(*) AS number_of_customer,
round(count(*)/(SELECT count(DISTINCT customer_id) FROM subscriptions)*100,0) AS percentage_of_nextplan
FROM cte_for_nextplan WHERE plan_id=0 AND nextplan IS NOT NULL
GROUP BY nextplan
ORDER BY nextplan;
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	nextplan	number_of_customer	percentage_of_nextplan			
▶	1	546	55			
	2	325	33			
	3	37	4			
	4	92	9			

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

```
SELECT plan_name, COUNT(customer_id) AS number_of_customer,
ROUND(COUNT(customer_id)/(SELECT COUNT(DISTINCT customer_id) FROM subscriptions)*100,0) AS percent_of_customer
FROM subscriptions s
JOIN plans p ON s.plan_id=p.plan_id
WHERE s.start_date<='2020-12-31'
GROUP BY plan_name;
```

Result Grid				Filter Rows:	Export:
	plan_name	number_of_customer	percent_of_customer		
▶	trial	1000	100		
	basic monthly	538	54		
	pro annual	195	20		
	churn	236	24		
	pro monthly	479	48		

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

```

SELECT COUNT(customer_id) AS upgraded_customers_count
FROM subscriptions
WHERE YEAR(start_date) = 2020
AND plan_id =(SELECT plan_id FROM plans WHERE plan_name='pro annual');

```

Result Grid		Filter Rows:	Export:
	upgraded_customers_count		
▶	195		

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(DATEDIFF(s.start_date, t.trial_start_date)),0) AS average_days_to_annual
FROM subscriptions s
JOIN (SELECT customer_id,
      MIN(start_date) AS trial_start_date
      FROM subscriptions
      GROUP BY customer_id) AS t ON s.customer_id = t.customer_id
JOIN plans p ON s.plan_id = p.plan_id
WHERE p.plan_name = 'pro annual';

```

Result Grid		Filter Rows:
	average_days_to_upgrade	
▶	105	

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

```

SELECT
  CASE
    WHEN DATEDIFF(s.start_date, t.trial_start_date) BETWEEN 0 AND 30 THEN '0-30 days'
    WHEN DATEDIFF(s.start_date, t.trial_start_date) BETWEEN 31 AND 60 THEN '31-60 days'
    WHEN DATEDIFF(s.start_date, t.trial_start_date) BETWEEN 61 AND 90 THEN '61-90 days'
    ELSE '> 90 days'
  END AS period,
  ROUND(AVG(DATEDIFF(s.start_date, t.trial_start_date)),0) AS average_days_to_annual
FROM subscriptions s
JOIN (SELECT customer_id, MIN(start_date) AS trial_start_date
      FROM subscriptions
      GROUP BY customer_id) AS t ON s.customer_id = t.customer_id
JOIN plans p ON s.plan_id = p.plan_id WHERE plan_name = 'pro annual'
GROUP BY period
ORDER BY period;

```

Result Grid			Filter Rows:	Export:
	period	average_days_to_annual		
▶	> 90 days	153		
	0-30 days	10		
	31-60 days	42		
	61-90 days	71		

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

```
WITH cte_for_nextplan AS(
    SELECT *,LEAD(plan_id,1) OVER(PARTITION BY customer_id) AS nextplan
    FROM subscriptions)
SELECT COUNT(customer_id) AS downgraded_customer_number
FROM cte_for_nextplan n
LEFT JOIN plans p ON p.plan_id=n.plan_id
WHERE p.plan_name='basic monthly'and p.plan_name='pro monthly' and year(n.start_date)='2020';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	downgraded_customer_number			
▶	0			