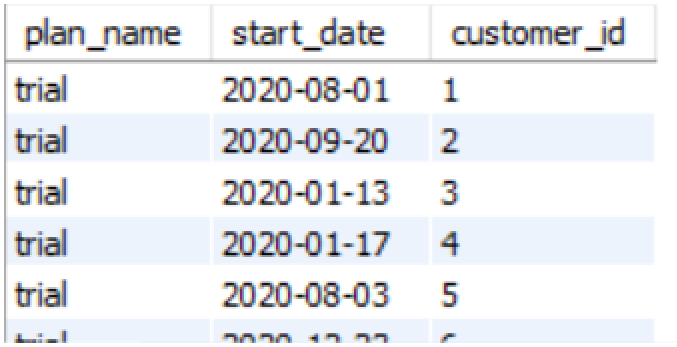
Case Study : Foodie-Fi

Data Analysis Solutions

Customer Journey:

```
select p.plan_name, s.start_date,s.customer_id

from subscriptions s

join plans p ON s.plan_id=p.plan_id;
```



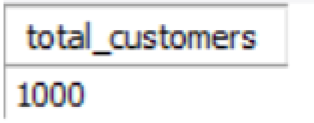
plan_name	start_date	customer_id
trial	2020-08-01	1
trial	2020-09-20	2
trial	2020-01-13	3
trial	2020-01-17	4
trial	2020-08-03	5

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

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

To find out how many customers Foodie-fi has, we can use COUNT and DISTINCT to return different values of customer_id.

Result:



total_customers
1000

Foodie-Fi has 1000 customers.

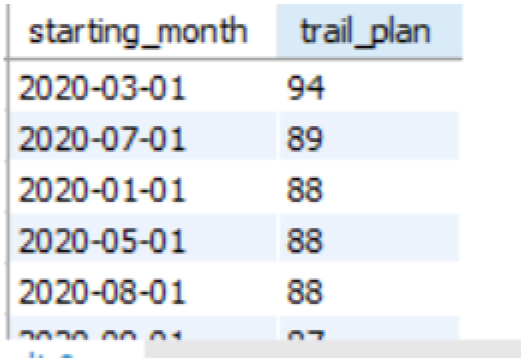
2. What is the monthly distribution of trial plan start_date values for our dataset - use the start of the month as the group by value?

```
select
        date_format(start_date, '%Y-%m-01') AS starting_month,
        count(plan_id) as trail_plan
From
        subscriptions
where
        plan_id=0
```

```
group by date_format(start_date, '%Y-%m-01')
order by starting_month;
```

we want to find the monthly distribution, we can use MONTH statement to extract the month of start_date.

Result:



starting_month	trail_plan
2020-03-01	94
2020-07-01	89
2020-01-01	88
2020-05-01	88
2020-08-01	88
2020-09-01	87

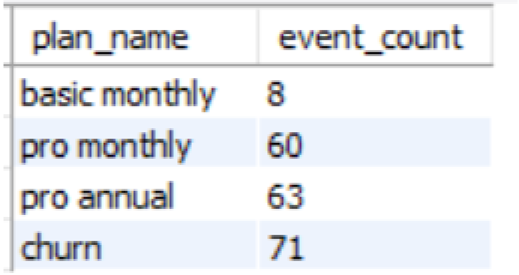
March has the biggest number of trial plan distribution.

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 plan_name, count(*) as event_count
from subscriptions
join plans on subscriptions.plan_id = plans.plan_id
where start_date > '2020-12-31'
group by plan_name;
```

plan_name and start_date are not in the same tables, we have to JOIN the tables. We can use JOIN clause to select records that have matching values in both tables. Then, filter the result with WHERE clause. In this case we filter the start_date after the year 2020.

Result:



plan_name	event_count
basic monthly	8
pro monthly	60
pro annual	63
churn	71

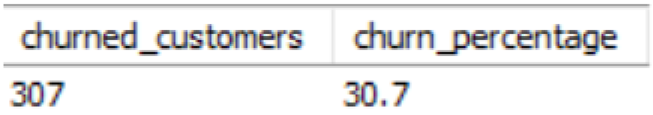
The number of customers who churned the plan is the biggest one after the **year 2020.**

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

```
SELECT
        COUNT(DISTINCT customer_id) AS churned_customers,
        ROUND((COUNT(DISTINCT customer_id) * 100.0 / (select COUNT(DISTINCT
customer_id) from subscriptions)), 1) AS churn_percentage
FROM
        subscriptions
WHERE plan_id = 4;
```

To calculate the percentage, total records multiplied by 100 and divided by total of customers. We use **ROUND** clause to rounded the percentage to 1 decimal place.

Result:



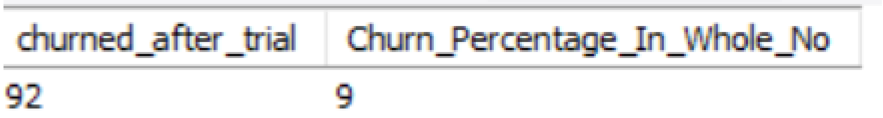
churned_customers	churn_percentage
307	30.7

It is **30.7%** of customers who have churned the plans.

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

```
SELECT
        COUNT(DISTINCT customer_id) AS churned_after_trial,
        Round(count(customer_id)/(select count(distinct customer_id) from subscriptions) *100)
As Churn_Percentage_In_Whole_No
From
        subscriptions
where
        plan_id= 4 AND day(start_date)<=8;
```

Result:



churned_after_trial	Churn_Percentage_In_Whole_No
92	9

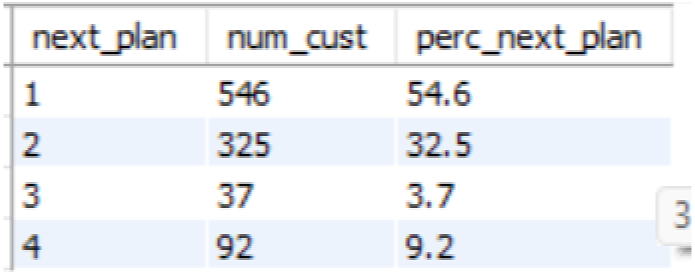
There are 92 customers who have churned straight after their initial free trial, which 9% of the customer base.

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

```
WITH cte_next_plan AS (

    SELECT

            *,

            LEAD(plan_id, 1) OVER(PARTITION BY customer_id ORDER BY plan_id) AS
next_plan

    FROM subscriptions)
```

```
SELECT
    next_plan,
    count(*) AS num_cust,
    ROUND(COUNT(*) * 100/(SELECT COUNT(DISTINCT customer_id) FROM
subscriptions),1) AS perc_next_plan
from cte_next_plan
where next_plan is not null and plan_id = 0
group by next_plan
order by next_plan;
```

Result:

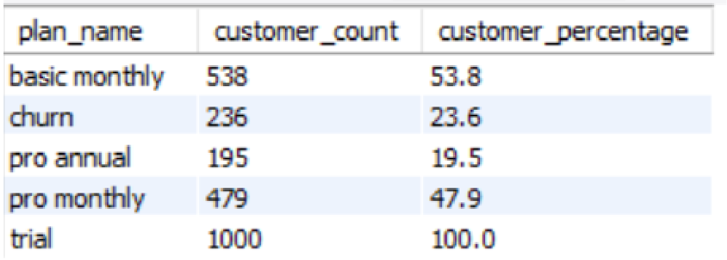


next_plan	num_cust	perc_next_plan
1	546	54.6
2	325	32.5
3	37	3.7
4	92	9.2

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

```
select
        plan_name, count(distinct customer_id) AS customer_count,
        ROUND(COUNT(DISTINCT customer_id) * 100.0 / (select count(distinct customer_id)
from subscriptions), 1) AS customer_percentage
From
        subscriptions
JOIN
        plans ON subscriptions.plan_id = plans.plan_id
WHERE
         start_date <= '2020-12-31'
GROUP BY plan_name;
```

Result:

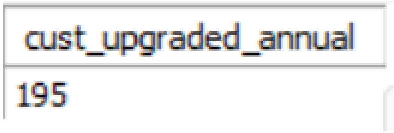


plan_name	customer_count	customer_percentage
basic monthly	538	53.8
churn	236	23.6
pro annual	195	19.5
pro monthly	479	47.9
trial	1000	100.0

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

```
select
        count(customer_id) as cust_upgraded_annual
from
        subscriptions
where plan_id=3 and year(start_date)=2020;
```

Result:



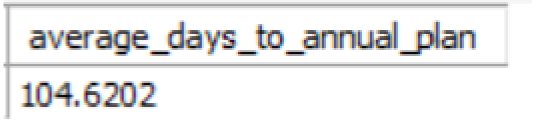
cust_upgraded_annual
195

195 customer upgraded for annual subscription.

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

```
select * from subscriptions;
with Annualplans as(
select customer_id , min(start_date) as annual_start_date from subscriptions where plan_id=3
group by customer_id
),
trialPlans as(
select customer_id , min(start_date) as trial_start_date from subscriptions where plan_id=0 group
by customer_id
)
select AVG(DATEDIFF(Annualplans.annual_start_date, trialPlans.trial_start_date))
AS average_days_to_annual_plan
FROM AnnualPlans
 join trialPlans on AnnualPlans.customer_id=trialPlans.customer_id;
```

Result:

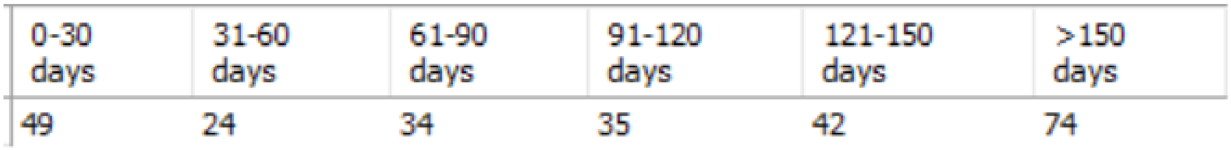


average_days_to_annual_plan
104.6202

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

```
WITH AnnualPlans AS (
   SELECT customer_id, MIN(start_date) AS annual_start_date
   FROM subscriptions
   WHERE plan_id = 3
   GROUP BY customer_id
),
TrialPlans AS (
  SELECT customer_id, MIN(start_date) AS trial_start_date
  FROM subscriptions
  WHERE plan_id = 0
  GROUP BY customer_id
),
DaysDifference AS (
  SELECT
    AnnualPlans.customer_id,
    DATEDIFF(AnnualPlans.annual_start_date, TrialPlans.trial_start_date) AS days_difference
  FROM
    AnnualPlans
  JOIN
    TrialPlans ON AnnualPlans.customer_id = TrialPlans.customer_id
)
SELECT
  SUM(CASE WHEN days_difference BETWEEN 0 AND 30 THEN 1 ELSE 0 END) AS "0-30 days",
  SUM(CASE WHEN days_difference BETWEEN 31 AND 60 THEN 1 ELSE 0 END) AS "31-60 days",
  SUM(CASE WHEN days_difference BETWEEN 61 AND 90 THEN 1 ELSE 0 END) AS "61-90 days",
  SUM(CASE WHEN days_difference BETWEEN 91 AND 120 THEN 1 ELSE 0 END) AS "91-120 days",
  SUM(CASE WHEN days_difference BETWEEN 121 AND 150 THEN 1 ELSE 0 END) AS "121-150 days",
  SUM(CASE WHEN days_difference > 150 THEN 1 ELSE 0 END) AS ">150 days"
FROM
  DaysDifference;
```

Result:

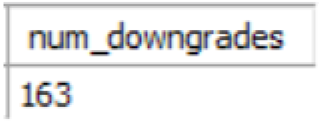


0-30 days	31-60 days	61-90 days	91-120 days	121-150 days	>150 days
49	24	34	35	42	74

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

```
select * from plans;

 select COUNT(DISTINCT customer_id) AS num_downgrades

from subscriptions

where plan_id =  2

and start_date >= '2020-01-01'
and start_date < '2021-01-01'
and customer_id in (
    select customer_id
    from subscriptions
    where plan_id = 1
    and start_date >= '2020-01-01'
    and start_date < '2021-01-01'
);
```

Result:



num_downgrades
163

163 person downgrade from a pro plan monthly subscription to basic monthly plan in 2020.