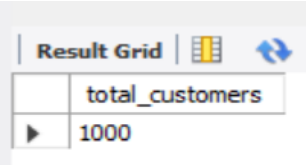
Foodie-Fi Subscription Analysis SQL Queries

Q1. How many customers has Foodie-Fi ever had?

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

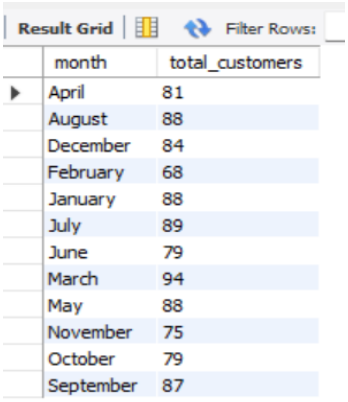


Result Grid

total_customers
1000

Q2. 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
  MONTHNAME(start_date) AS month,
  COUNT(customer_id) AS total_customers
FROM subscriptions
WHERE plan_id = 0
GROUP BY MONTHNAME(start_date)
ORDER BY MONTHNAME(start_date);
```

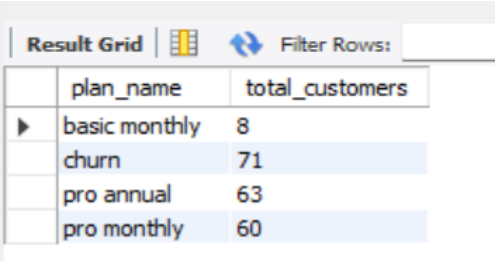


Result Grid | Filter Rows:

month	total_customers
April	81
August	88
December	84
February	68
January	88
July	89
June	79
March	94
May	88
November	75
October	79
September	87

Q3. 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(s.customer_id) AS total_customers
FROM plans AS p
JOIN subscriptions AS s ON p.plan_id = s.plan_id
WHERE YEAR(start_date) > 2020
GROUP BY p.plan_name
ORDER BY p.plan_name;
```

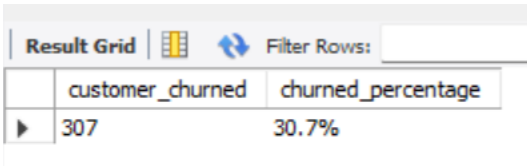

Result Grid | Filter Rows:

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

Q4. What is the customer count and percentage of customers who have churned rounded to 1 decimal place?

```
SELECT
  customer_churned,
  CONCAT(ROUND((customer_churned / total_customer) * 100, 1), '%') AS
churned_percentage
FROM (
   SELECT
      (SELECT COUNT(s.customer_id)
       FROM plans AS p
       JOIN subscriptions AS s ON p.plan_id = s.plan_id
```

```
        WHERE p.plan_id = 4) AS customer_churned,
       (SELECT COUNT(DISTINCT s.customer_id)
        FROM plans AS p
        JOIN subscriptions AS s ON p.plan_id = s.plan_id) AS total_customer
) AS x;
```



Result Grid | Filter Rows:

	customer_churned	churned_percentage
▶	307	30.7%

Q5. How many customers have churned straight after their initial free trial. what percentage is this rounded to the nearest whole number?

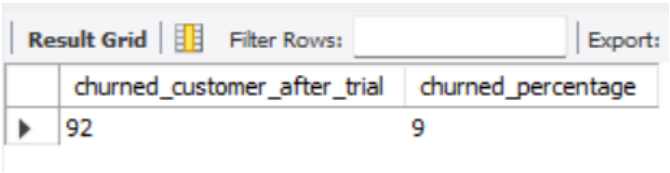
```
# With Join
WITH churn_customer AS (
   SELECT *
   FROM (
      SELECT s.*, p.plan_name,
           ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY start_date) AS rn
      FROM subscriptions AS s
      JOIN plans AS p ON s.plan_id = p.plan_id
   ) AS t
   WHERE rn = 2 AND plan_name = "churn"
)
SELECT
  COUNT(*) AS churned_customer_after_trial,
  CONCAT(ROUND(COUNT(*) / (SELECT COUNT(DISTINCT customer_id) FROM
subscriptions) * 100, 0), '%') AS churned_percentage
```

```
FROM churn_customer;


# Without Join

WITH CTE AS (
   SELECT *
   FROM (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY start_date)
AS rn
    FROM subscriptions ) AS t
    WHERE rn = 2 AND plan_id =4)


SELECT COUNT(*) AS churned_customer_after_trial,
ROUND(COUNT(*) / (SELECT COUNT(DISTINCT customer_id) FROM subscriptions)
*100,0) AS churned_percentage
FROM CTE;
```



Result Grid | Filter Rows: | Export:

churned_customer_after_trial	churned_percentage
92	9

Q6. What is the number and percentage of customer plans after their initial free trial?

```
WITH CTE AS (
   SELECT
      customer_id, plan_name,
      ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY start_date) AS rn
   FROM subscriptions AS s
   JOIN plans AS p ON s.plan_id = p.plan_id
```

```
)
SELECT
  plan_name,
  COUNT(*) AS customer_after_trial,
  CONCAT(ROUND(COUNT(*) / (SELECT COUNT(DISTINCT customer_id) FROM CTE) * 100, 2), '%') AS percentage
FROM CTE
WHERE rn = 2
GROUP BY plan_name;
```

Result Grid | Filter Rows: | Export:

plan_name	customer_after_trial	percentage
basic monthly	546	54.60%
pro annual	37	3.70%
pro monthly	325	32.50%
churn	92	9.20%

Q7. What is the customer count and percentage breakdown of all 5 plan_name values at 2020-12-31?

```
WITH CTE AS (
   SELECT *,
        ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY start_date DESC) AS rn
   FROM subscriptions
   WHERE start_date <= '2020-12-31'
)
SELECT
  p.plan_name,
  COUNT(CTE.customer_id) AS customer,
```

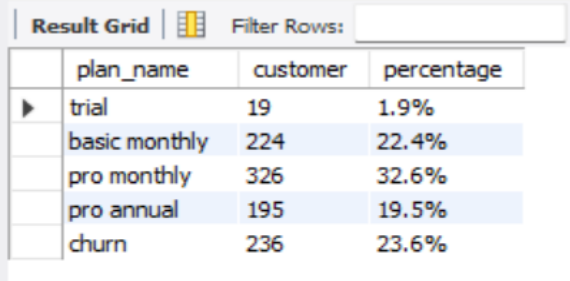
```
  CONCAT(ROUND(COUNT(CTE.customer_id) / (SELECT COUNT(DISTINCT customer_id)
FROM CTE) * 100, 1), '%') AS percentage

FROM CTE

JOIN plans AS p ON CTE.plan_id = p.plan_id

WHERE rn = 1

GROUP BY p.plan_name;
```



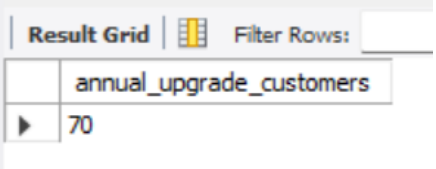
Result Grid | Filter Rows:

plan_name	customer	percentage
trial	19	1.9%
basic monthly	224	22.4%
pro monthly	326	32.6%
pro annual	195	19.5%
churn	236	23.6%

Q8. How many customers have upgraded to an annual plan in 2020?

```
WITH monthly_customers AS (
   SELECT
      customer_id, start_date
   FROM subscriptions AS S
   WHERE YEAR(start_date) = 2020 AND plan_id IN (2)
),
annual_customers AS (
   SELECT
      customer_id, start_date
   FROM subscriptions AS S
   WHERE YEAR(start_date) = 2020 AND plan_id = 3
)
SELECT
```

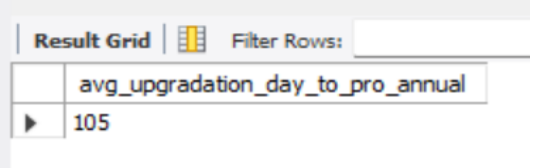
```
    COUNT(DISTINCT A.customer_id) AS annual_upgrade_customers
FROM monthly_customers AS M
INNER JOIN annual_customers AS A
    ON M.customer_id = A.customer_id AND M.start_date < A.start_date;
```



Q9. How many days on average does it take for a customer to an annual plan from the day they join Foodie-Fi?

```
WITH first_plan AS (
    SELECT *,
        ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY start_date) AS rn
    FROM subscriptions
    WHERE plan_id = 0
),
pro_annual_plan AS (
    SELECT *,
        ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY start_date) AS rn
    FROM subscriptions
    WHERE plan_id = 3
)
SELECT ROUND(AVG(DATEDIFF(p.start_date, f.start_date)), 0) AS
avg_upgradation_day_to_pro_annual
FROM first_plan AS f
JOIN pro_annual_plan AS p ON f.customer_id = p.customer_id
```

```
WHERE f.start_date < p.start_date;
```

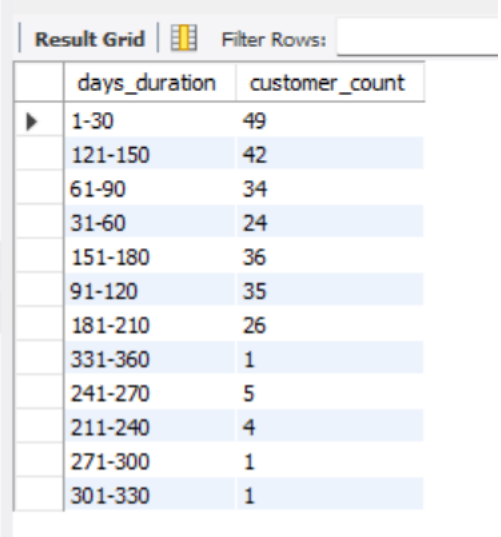


avg_upgradation_day_to_pro_annual
105

Q10. Can you further breakdown this average value into 30 day periods (i.e. 1-30 days, 31-60 days etc)

```
WITH Trial AS (
   SELECT customer_id, start_date AS trial_start
   FROM subscriptions
   WHERE plan_id = 0
),
Annual AS (
   SELECT customer_id, start_date AS annual_start
   FROM subscriptions
   WHERE plan_id = 3
)
SELECT
   CONCAT(
     FLOOR((DATEDIFF(A.annual_start, T.trial_start) - 1) / 30) * 30 + 1,
     '-',
     FLOOR((DATEDIFF(A.annual_start, T.trial_start) - 1) / 30) * 30 + 30
   ) AS days_duration,
   COUNT(T.customer_id) AS customer_count
FROM Trial AS T
INNER JOIN Annual AS A ON T.customer_id = A.customer_id
```

GROUP BY days_duration;



Result Grid | Filter Rows:

days_duration	customer_count
1-30	49
121-150	42
61-90	34
31-60	24
151-180	36
91-120	35
181-210	26
331-360	1
241-270	5
211-240	4
271-300	1
301-330	1

Q11. Can you further breakdown this average value into 30 day periods (i.e. 1-30 days, 31-60 days etc)

```
WITH pro_monthly_plan AS (
   SELECT *
   FROM subscriptions
   WHERE YEAR(start_date) = 2020 AND plan_id = 2
),
basic_monthly_plan AS (
   SELECT *
   FROM subscriptions
   WHERE YEAR(start_date) = 2020 AND plan_id = 1
)
SELECT COUNT(DISTINCT b.customer_id) AS customers_downgraded
FROM pro_monthly_plan AS p
JOIN basic_monthly_plan AS b
   ON p.customer_id = b.customer_id
```

WHERE p.start_date < b.start_date;