SQL PORTFOLIO PROJECT

FOODIE – FI CASE STUDY

BY HANIFA IBRAHIM

GITHUB: https://github.com/Hanifa-Ibrahim/Foodie-Fi

MEDIUM: https://medium.com/@honeyfa2002/foodie-fi-case-study-1fd3fa98db53

QUESTION: 1

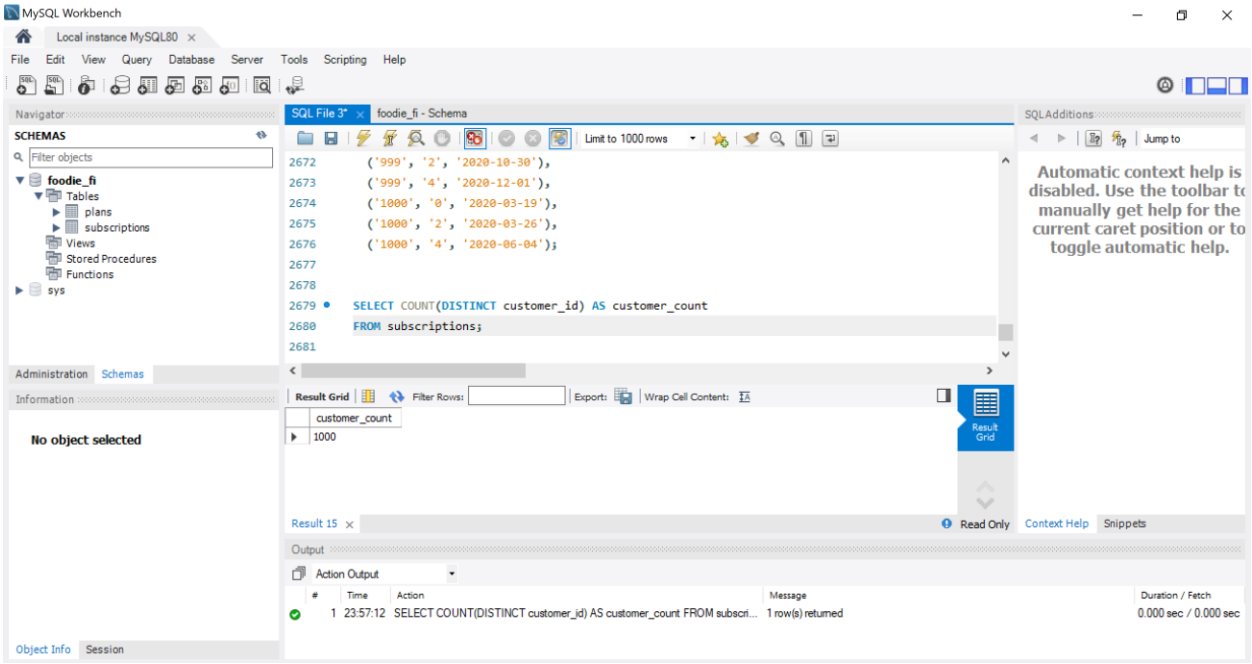
How many customers Foodie-Fi ever had?

QUERY

```
SELECT COUNT(DISTINCT customer_id) AS customer_count

FROM subscriptions;
```

OUTPUT



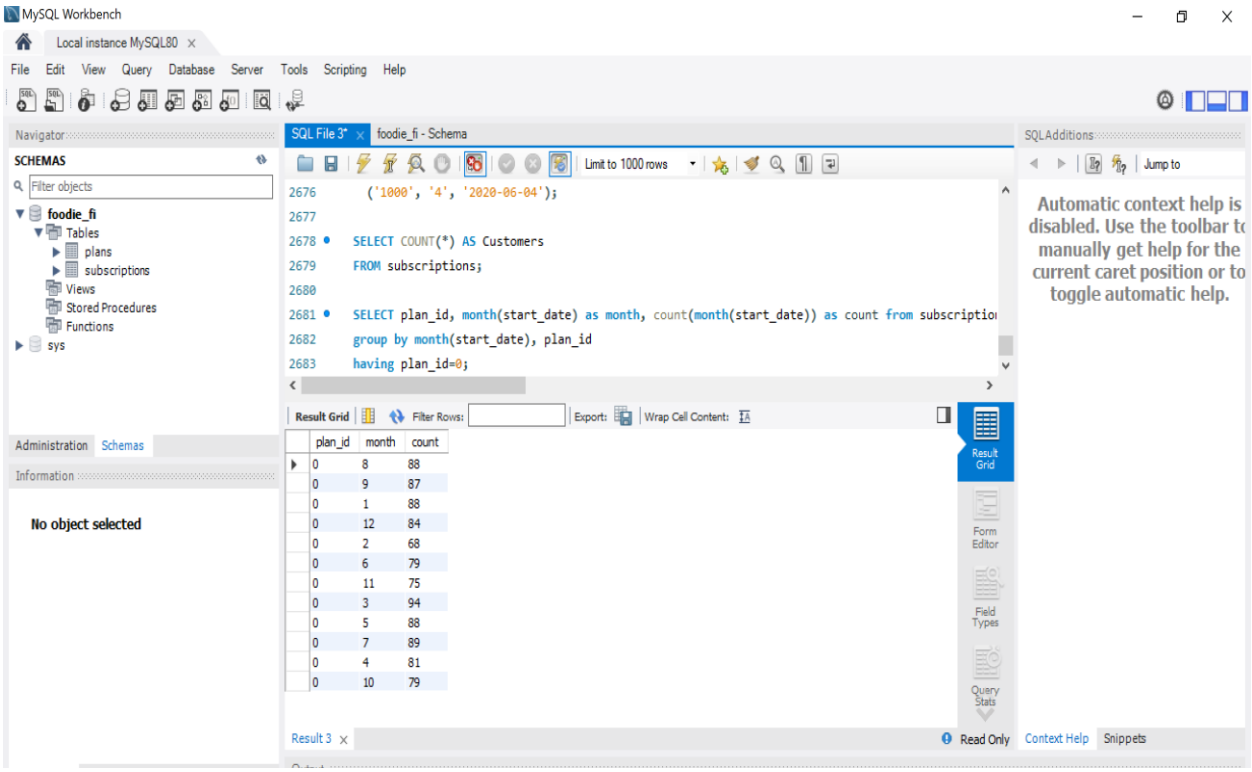
QUESTION: 2

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.

QUERY

```
SELECT plan_id, month(start_date) as month, count(month(start_date)) as count from subscriptions
group by month(start_date), plan_id
having plan_id=0;
```

OUTPUT



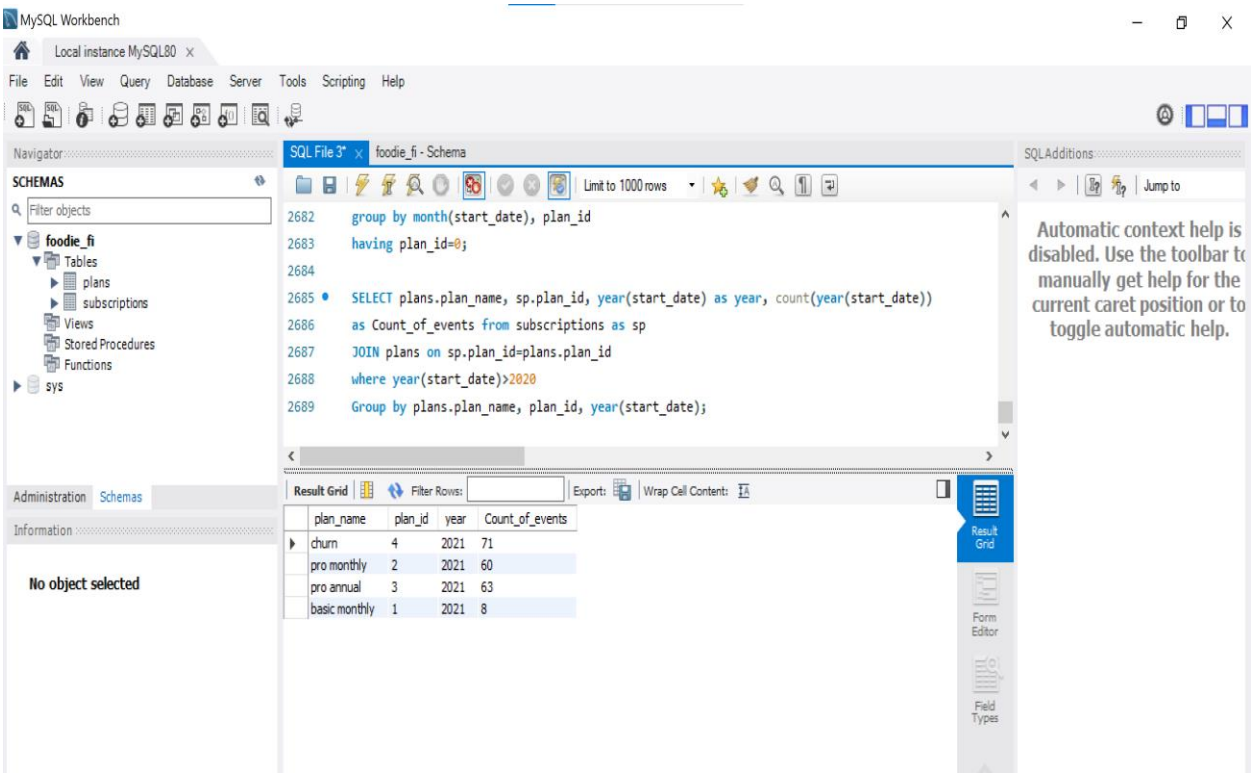
QUESTION: 3

What plan start_date values occur after the year 2020 for our dataset? Show the breakdown by count of events for each plan_name.

QUERY

```
SELECT plans.plan_name, sp.plan_id, year(start_date) as year, count(year(start_date))
as Count_of_events from subscriptions as sp
JOIN plans on sp.plan_id=plans.plan_id
where year(start_date)>2020
Group by plans.plan_name, plan_id, year(start_date);
```

OUTPUT



QUESTION: 4

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

QUERY

select plan_name, count((select count(distinct customer_id) from subscriptions)) as count_of_churned,

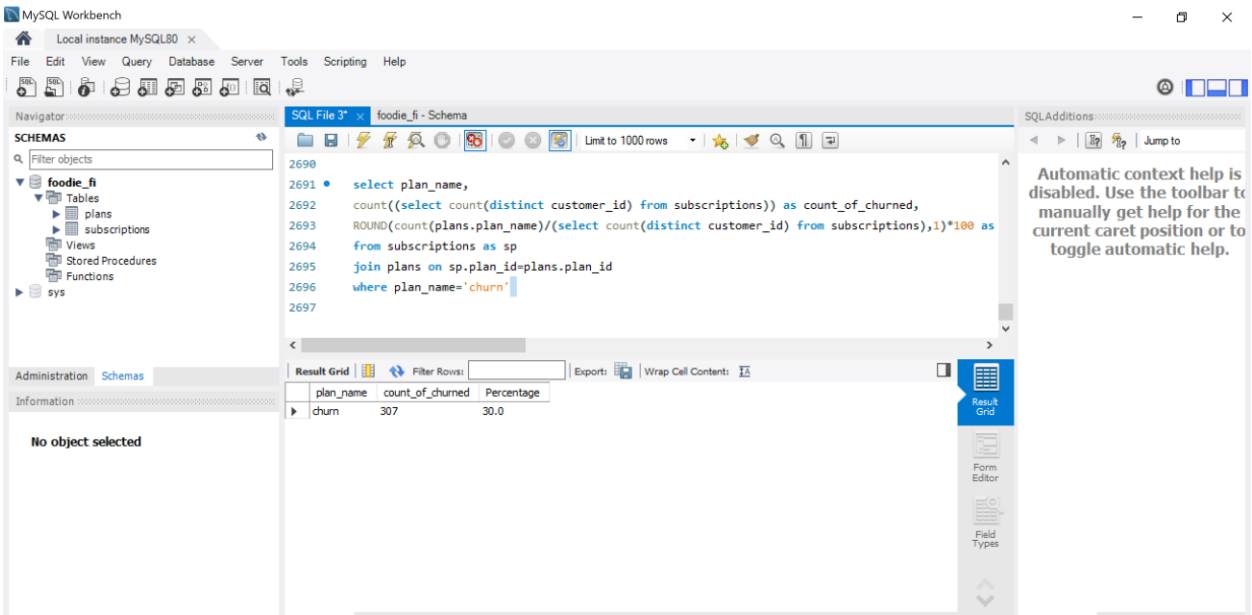
ROUND(count(plans.plan_name) / (select count(distinct customer_id) from subscriptions),1)*100 as Percentage

from subscriptions as sp

join plans on sp.plan_id=plans.plan_id

where plan_name='churn'

OUTPUT



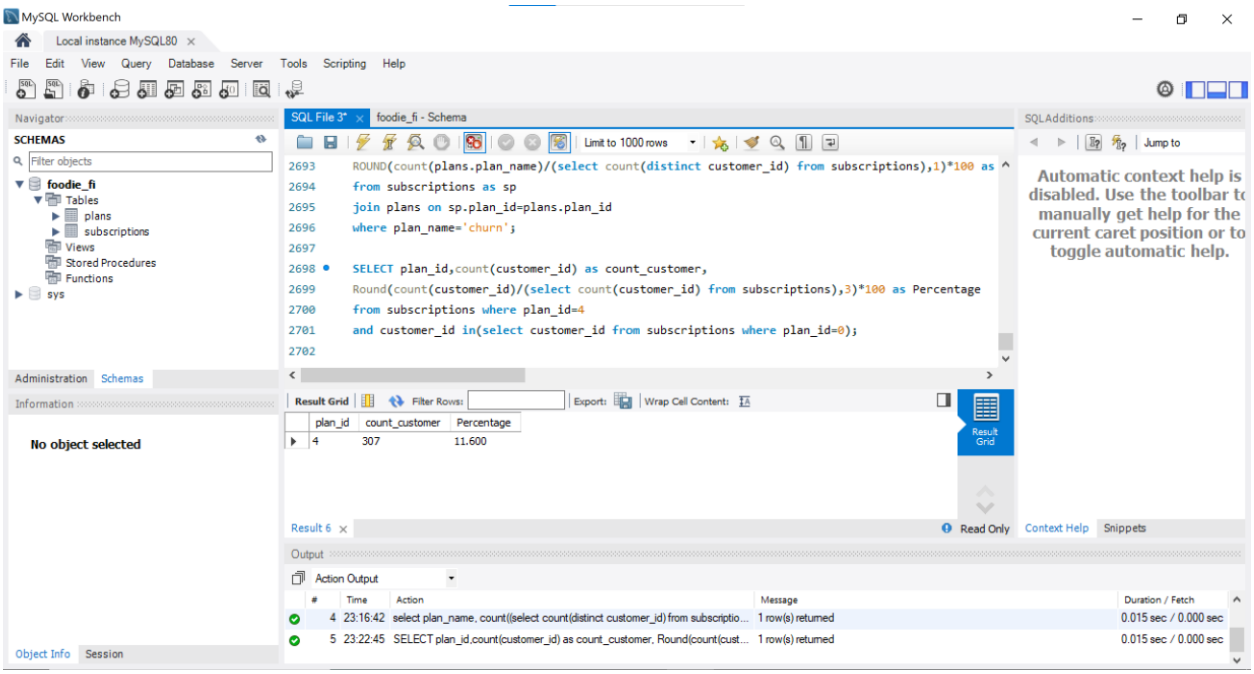
QUESTION: 5

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

QUERY

```
SELECT plan_id,count(customer_id) as count_customer,
Round(count(customer_id)/(select count(customer_id) from subscriptions),3)*100 as
Percentage
from subscriptions where plan_id=4
and customer_id in(select customer_id from subscriptions where plan_id=0);
```

OUTPUT



QUESTION: 6

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

QUERY

```
WITH CustomerFirstPaidPlan AS (
   SELECT customer_id, MIN(plan_id) AS first_paid_plan_id
   FROM subscriptions
   WHERE plan_id != 0 -- Exclude free trial
   GROUP BY customer_id
)
SELECT p.plan_name,
      COUNT(DISTINCT c.customer_id) AS customer_count,
      ROUND(COUNT(DISTINCT c.customer_id) * 100.0 / (SELECT COUNT(DISTINCT customer_id) FROM
subscriptions)) AS percentage
FROM CustomerFirstPaidPlan c
JOIN plans p ON c.first_paid_plan_id = p.plan_id
GROUP BY p.plan_name;
```

OUTPUT

```
WITH CustomerFirstPaidPlan AS (
    SELECT customer_id, MIN(plan_id) AS first_paid_plan_id
    FROM subscriptions
    WHERE plan_id != 0 -- Exclude free trial
    GROUP BY customer_id
)
SELECT p.plan_name,
       COUNT(DISTINCT c.customer_id) AS customer_count,
       ROUND(COUNT(DISTINCT c.customer_id) * 100.0 / (SELECT COUNT(DISTINCT customer_id) FROM subscriptions)) AS pe
FROM CustomerFirstPaidPlan c
JOIN plans p ON c.first_paid_plan_id = p.plan_id
GROUP BY p.plan_name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

plan_name	customer_count	percentage
basic monthly	546	55
churn	92	9
pro annual	37	4
pro monthly	325	33

QUESTION: 7

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

QUERY

```
WITH cte1 AS (
SELECT *, RANK() OVER(PARTITION BY customer_id ORDER BY start_date DESC) AS ranking
FROM subscriptions
WHERE start_date <= '2020-12-31'
),
cte2 AS (
SELECT p.plan_name, COUNT(cte1.plan_id) AS num_plans
FROM cte1 INNER JOIN plans p
ON cte1.plan_id = p.plan_id
WHERE cte1.ranking = 1
GROUP BY p.plan_name
ORDER BY num_plans DESC
)
SELECT plan_name, num_plans,
ROUND((num_plans/(SUM(num_plans) OVER())*100), 2) AS pct_plans
FROM cte2;
```

OUTPUT

```
WITH cte1 AS (
SELECT *, RANK() OVER(PARTITION BY customer_id ORDER BY start_date DESC) AS ranking
FROM subscriptions
WHERE start_date <= '2020-12-31'
),
cte2 AS (
SELECT p.plan_name, COUNT(cte1.plan_id) AS num_plans
FROM cte1 INNER JOIN plans p
ON cte1.plan_id = p.plan_id
WHERE cte1.ranking = 1
GROUP BY p.plan_name
ORDER BY num_plans DESC
)
SELECT plan_name, num_plans,
ROUND((num_plans/(SUM(num_plans) OVER())*100), 2) AS pct_plans
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

plan_name	num_plans	pct_plans
pro monthly	1304	32.60
churn	944	23.60
basic monthly	896	22.40
pro annual	780	19.50
trial	76	1.90

QUESTION: 8

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

QUERY

```
SELECT COUNT(DISTINCT s.customer_id) AS num_customers
FROM subscriptions s INNER JOIN plans p
ON s.plan_id = p.plan_id
WHERE p.plan_name = "pro annual" AND YEAR(s.start_date) = 2020;
```

OUTPUT

```
SELECT COUNT(DISTINCT s.customer_id) AS num_customers
FROM subscriptions s INNER JOIN plans p
ON s.plan_id = p.plan_id
WHERE p.plan_name = "pro annual" AND YEAR(s.start_date) = 2020;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

num_customers
195

QUESTION: 9

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

QUERY

```
WITH cte1 AS (
SELECT * FROM subscriptions
WHERE plan_id = 0
),
cte2 AS (
SELECT * FROM subscriptions
WHERE plan_id = 3
)
SELECT ROUND(AVG(DATEDIFF(cte2.start_date, cte1.start_date)), 1) AS avg_days
FROM cte1 JOIN cte2
ON cte1.customer_id = cte2.customer_id;
```

OUTPUT

```
WITH cte1 AS (
SELECT * FROM subscriptions
WHERE plan_id = 0
),
cte2 AS (
SELECT * FROM subscriptions
WHERE plan_id = 3
)
SELECT ROUND(AVG(DATEDIFF(cte2.start_date, cte1.start_date)), 1) AS avg_days
FROM cte1 JOIN cte2
ON cte1.customer_id = cte2.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_days
104.6

QUESTION: 10

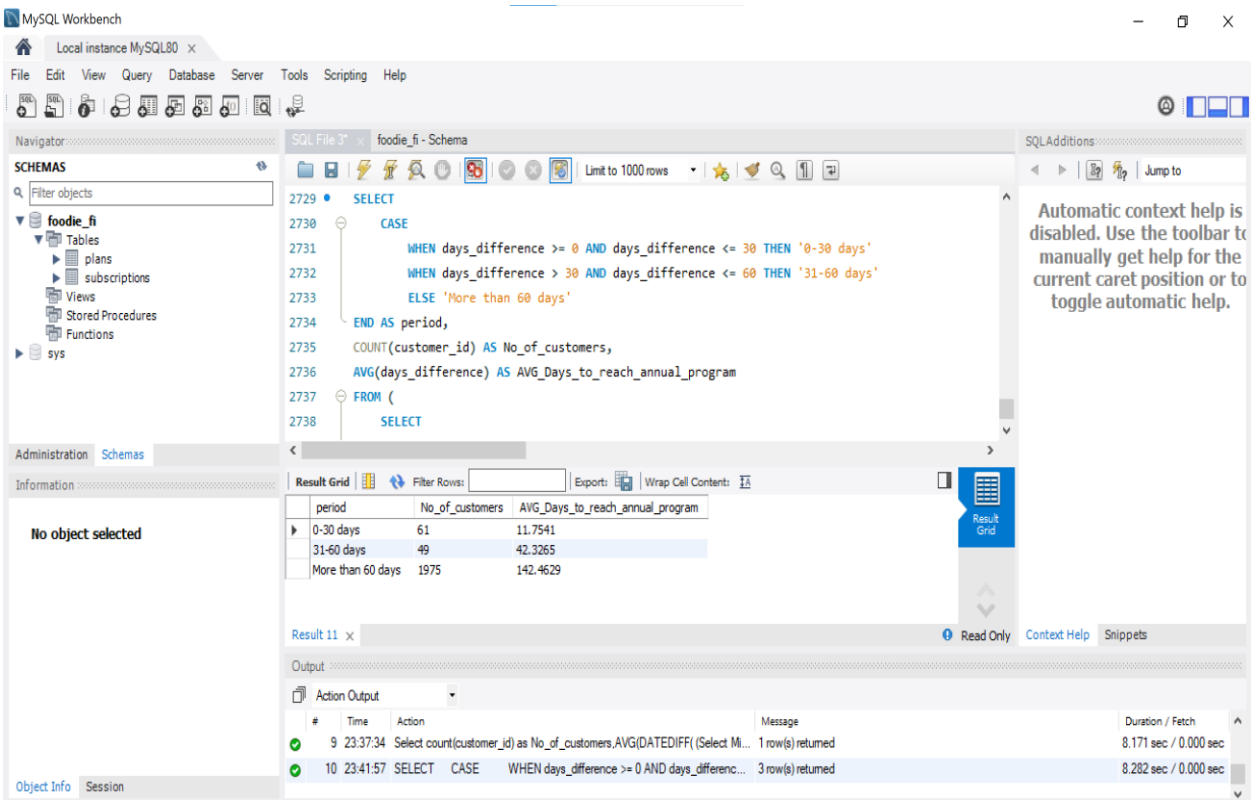
Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)

QUERY

```
SELECT
  CASE
    WHEN days_difference >= 0 AND days_difference <= 30 THEN '0-30 days'
    WHEN days_difference > 30 AND days_difference <= 60 THEN '31-60 days'
    ELSE 'More than 60 days'
END AS period,
COUNT(customer_id) AS No_of_customers,
AVG(days_difference) AS AVG_Days_to_reach_annual_program
```

```
FROM (
   SELECT
      s2.customer_id,
      DATEDIFF(
         (SELECT MIN(start_date) FROM subscriptions AS s1 WHERE s1.customer_id = s2.customer_id AND plan_id = 3),
 (SELECT MIN(start_date) FROM subscriptions AS s3 WHERE s3.customer_id = s2.customer_id)) AS days_difference
FROM subscriptions AS s2 WHERE plan_id ! = 3 AND plan_id ! = 4) AS differences
GROUP BY period
ORDER BY period;
```

OUTPUT



QUESTION: 11

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

QUERY

select plan_id, count(distinct customer_id) as downgraded_from_annual_to_basic from subscriptions

where plan_id =1

and customer_id in

(select distinct customer_id from subscriptions where plan_id=2 and year(start_date) = 2020);

OUTPUT