**SQL PORTFOLIO PROJECT**

**FOODIE – FI CASE STUDY**

**BY AMNA BIBI**

**Medium :**

**https://medium.com/@amnashah1866/unveiling-the-recipe-for-success-a-data-driven-journey-with-foodie-fi-1cf93a01d324**

**Github:**

**https://github.com/amnaashah/sql-case-study**

**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.

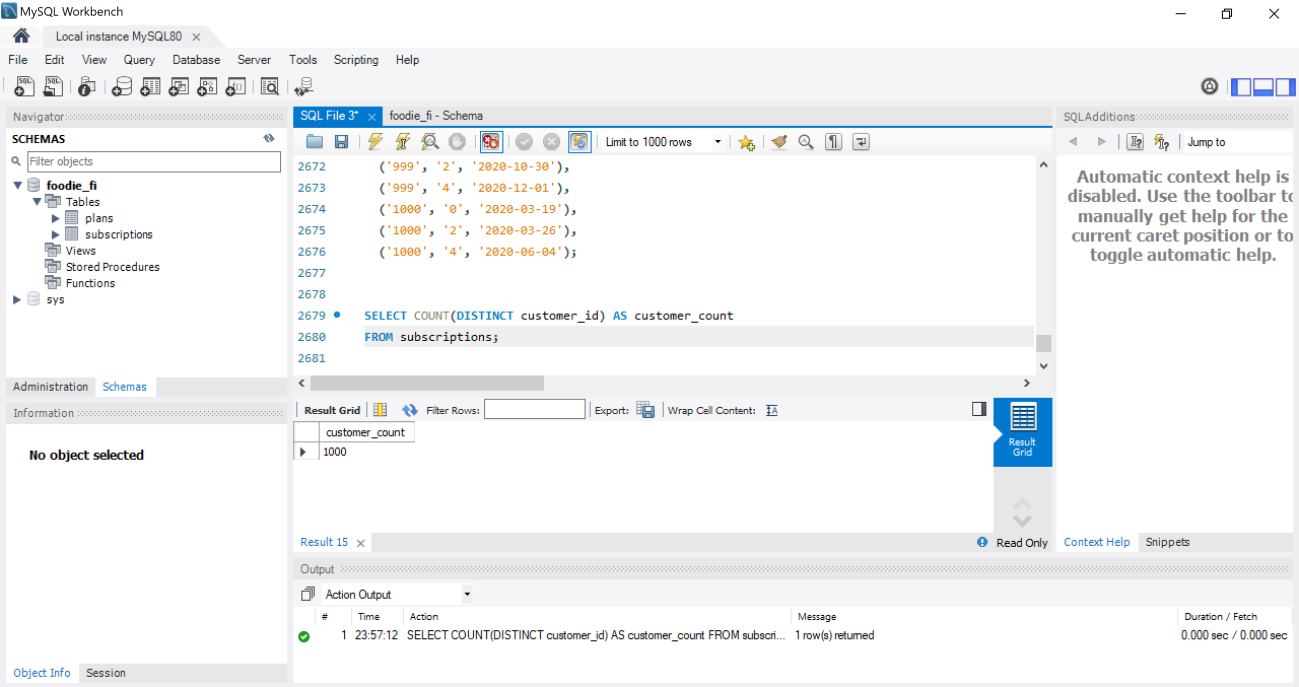
**QUESTION: 1**

**How many customers Foodie-Fi ever had?**

**QUERY:**

**SELECT COUNT(DISTINCT customer\_id) AS customer\_count**

**FROM subscriptions;**



**QUESTION: 2**

**What is the monthly distribution of trial plan start\_date values for our dataset - use the start of**

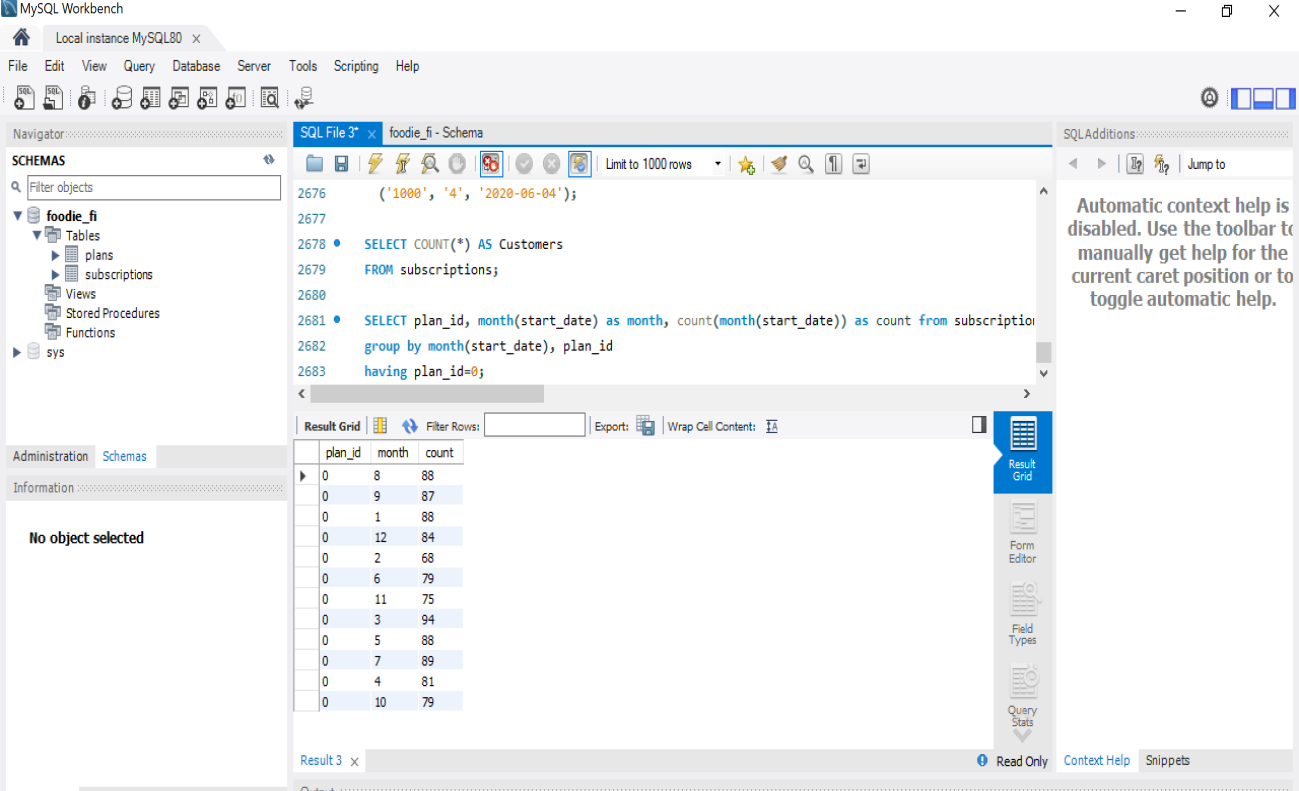
**the month as the group by value.**

**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;**

****

**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**

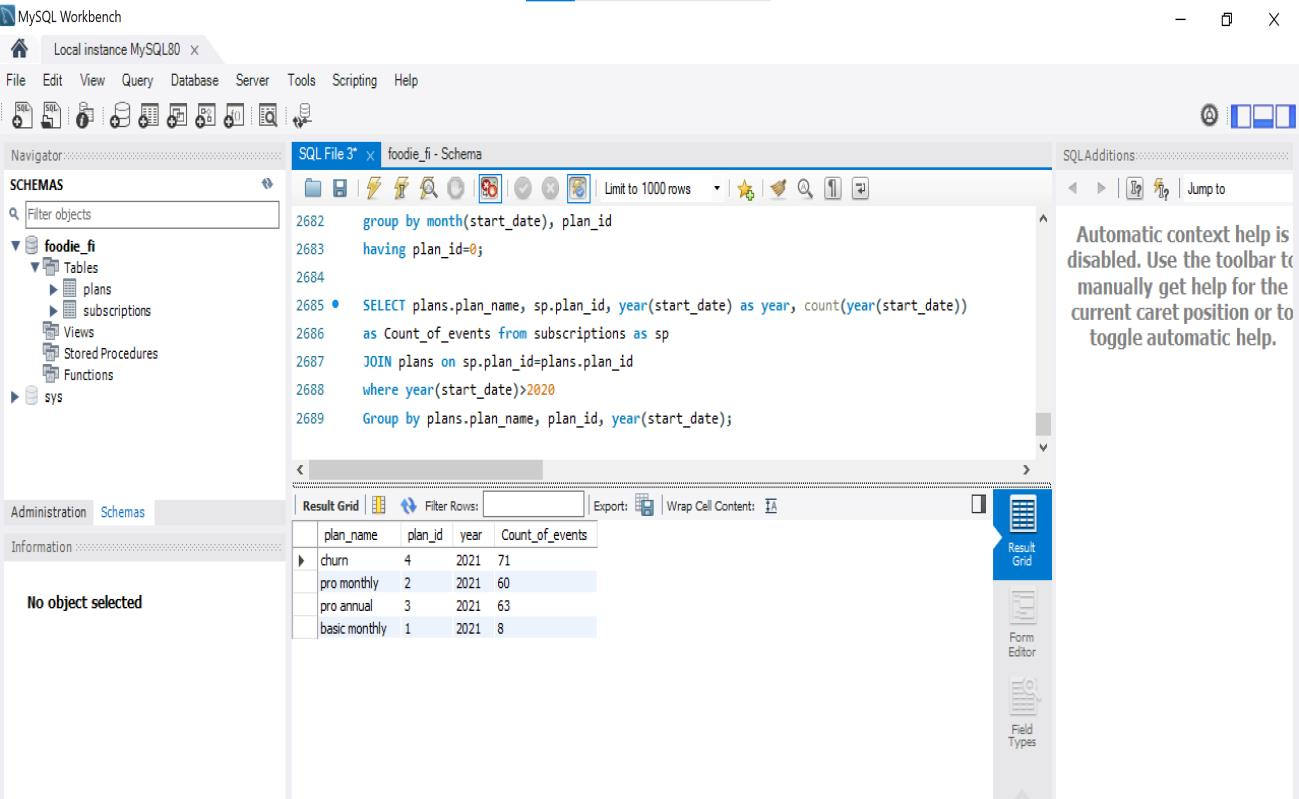
**SELECT plans.plan\_name, sp.plan\_id, year(start\_date) as year, count(year(start\_date))**

**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);**

****

**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,**

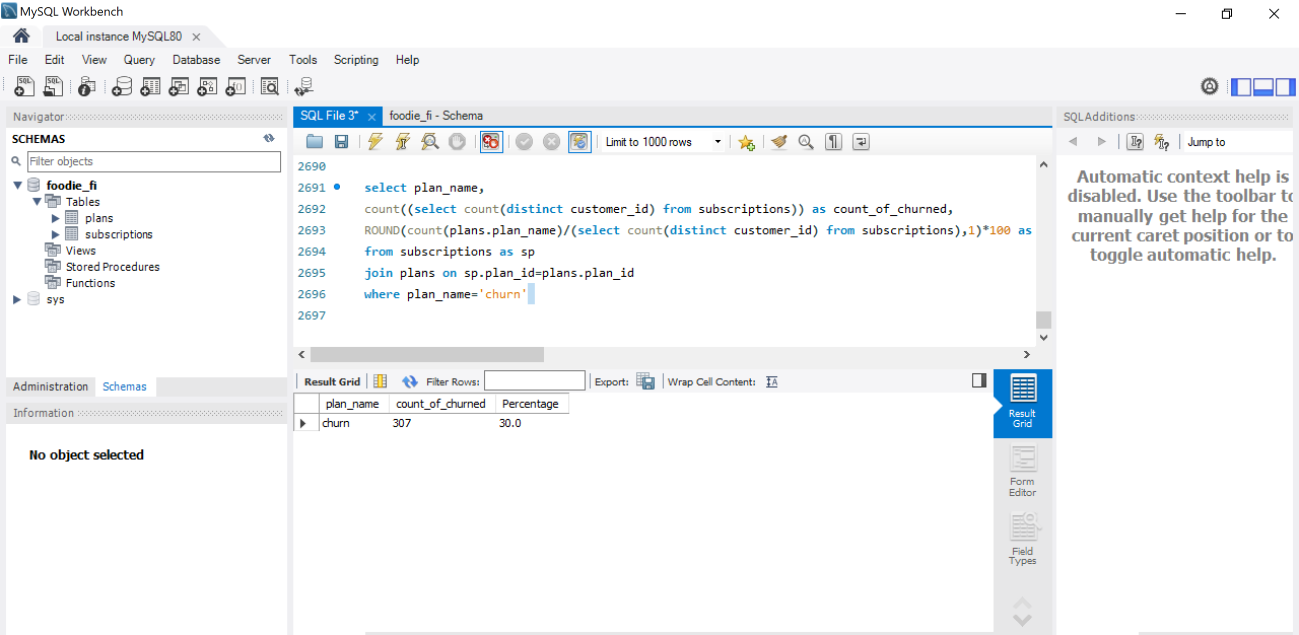
**ROUND(count(plans.plan\_name) / (select count(distinct customer\_id) from**

**subscriptions),1)\*100 as Percentage**

**from subscriptions as sp**

**join plans on sp.plan\_id=plans.plan\_id**

**where plan\_name='churn'**



**QUESTION: 5**

**How many customers have churned straight after their initial free trial - what percentage is this**

**rounded to the nearest whole number?**

**QUERY**

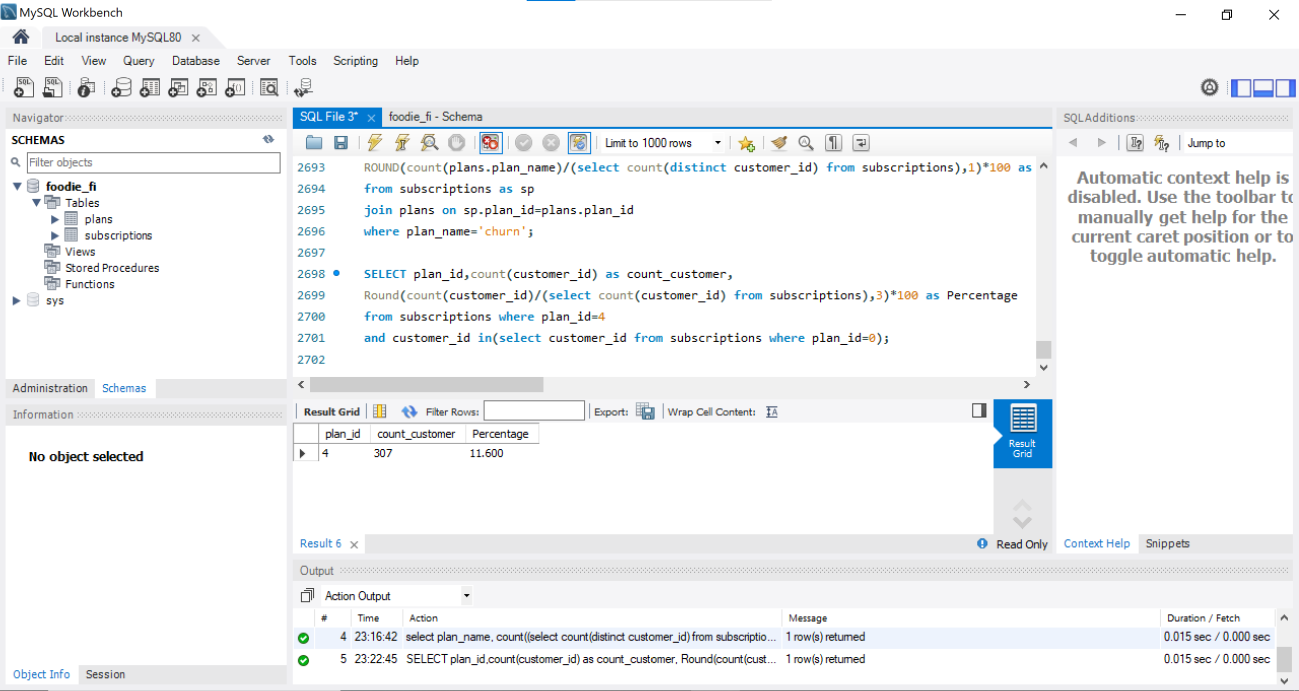
**SELECT plan\_id,count(customer\_id) as count\_customer,**

**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);**



**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,**

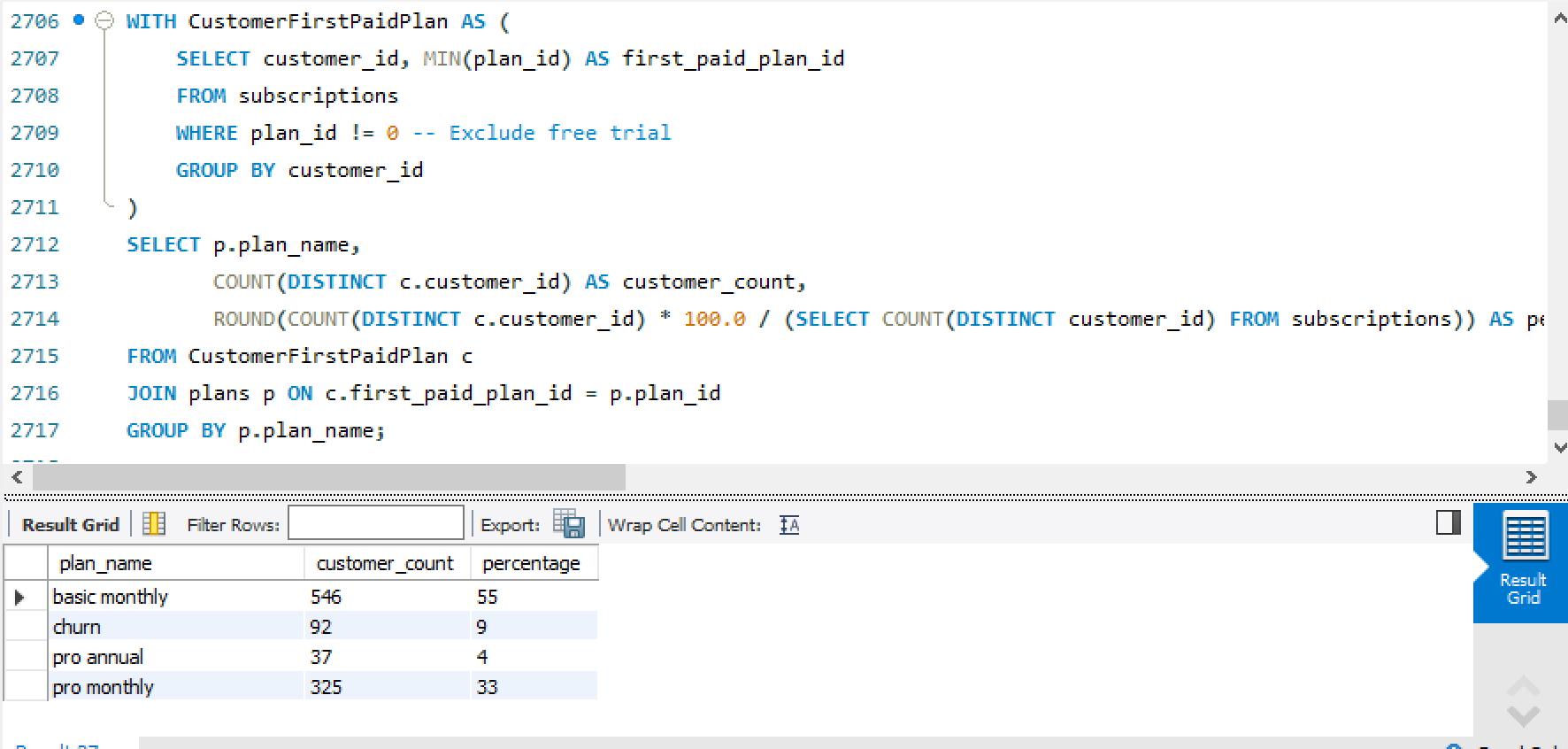
**ROUND(COUNT(DISTINCT c.customer\_id) \* 100.0 / (SELECT COUNT(DISTINCT customer\_id) FROM**

**subscriptions)) AS percentage**

**FROM CustomerFirstPaidPlan c**

**JOIN plans p ON c.first\_paid\_plan\_id = p.plan\_id**

**GROUP BY p.plan\_name;**



**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**

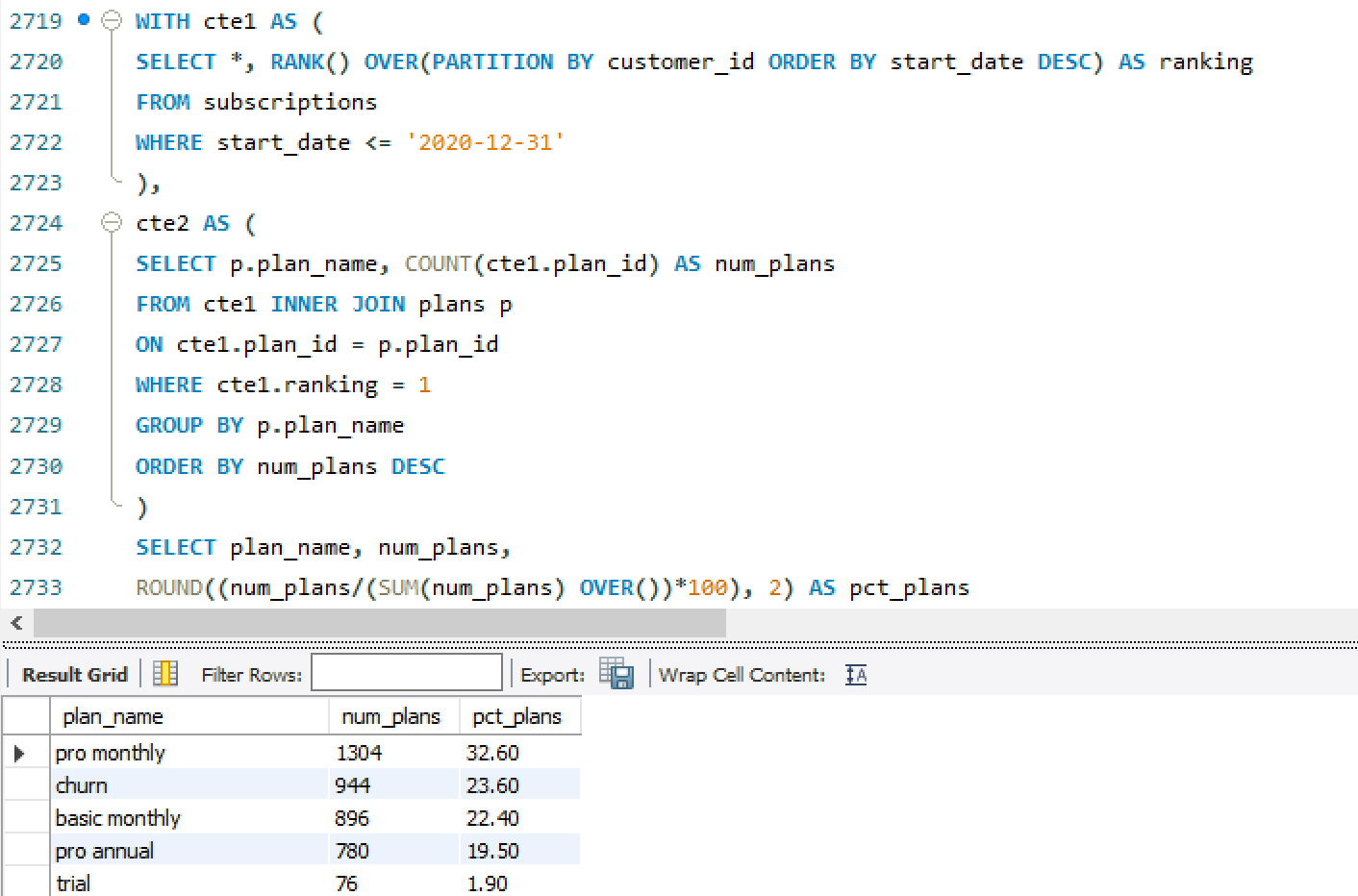
**ORDER BY num\_plans DESC**

**)**

**SELECT plan\_name, num\_plans,**

**ROUND((num\_plans/(SUM(num\_plans) OVER())\*100), 2) AS pct\_plans**

**FROM cte2**



**QUESTION: 8**

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

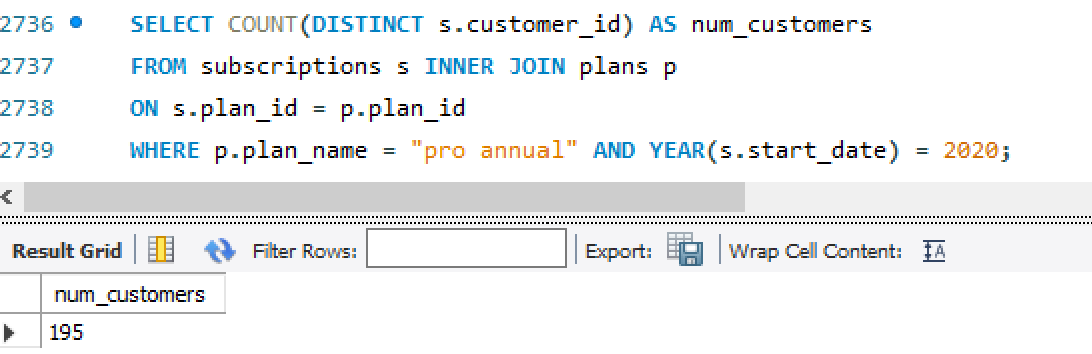
**QUERY**

**SELECT COUNT(DISTINCT s.customer\_id) AS num\_customers**

**FROM subscriptions sINNER JOIN plans p**

**ON s.plan\_id = p.plan\_id**

**WHERE p.plan\_name = "pro annual" AND YEAR(s.start\_date) = 2020;**



**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**

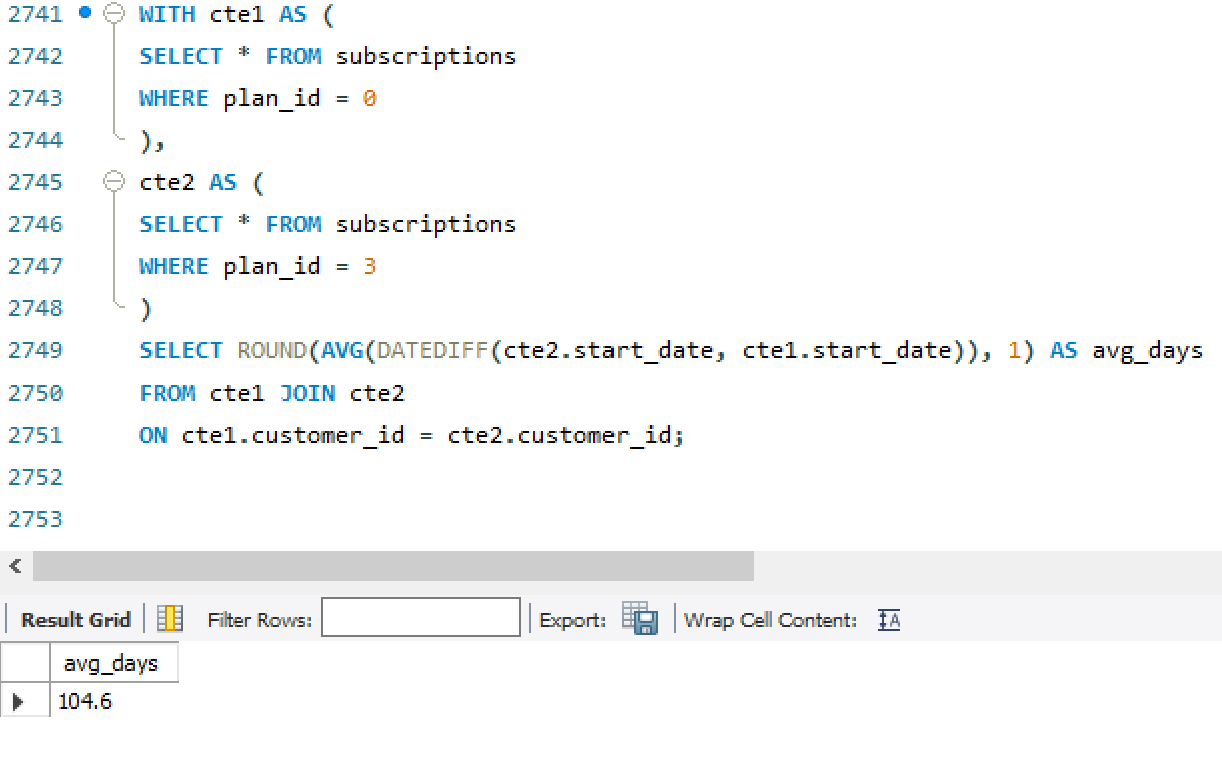
**WHERE plan\_id = 3**

**)**

**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;**



**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),**

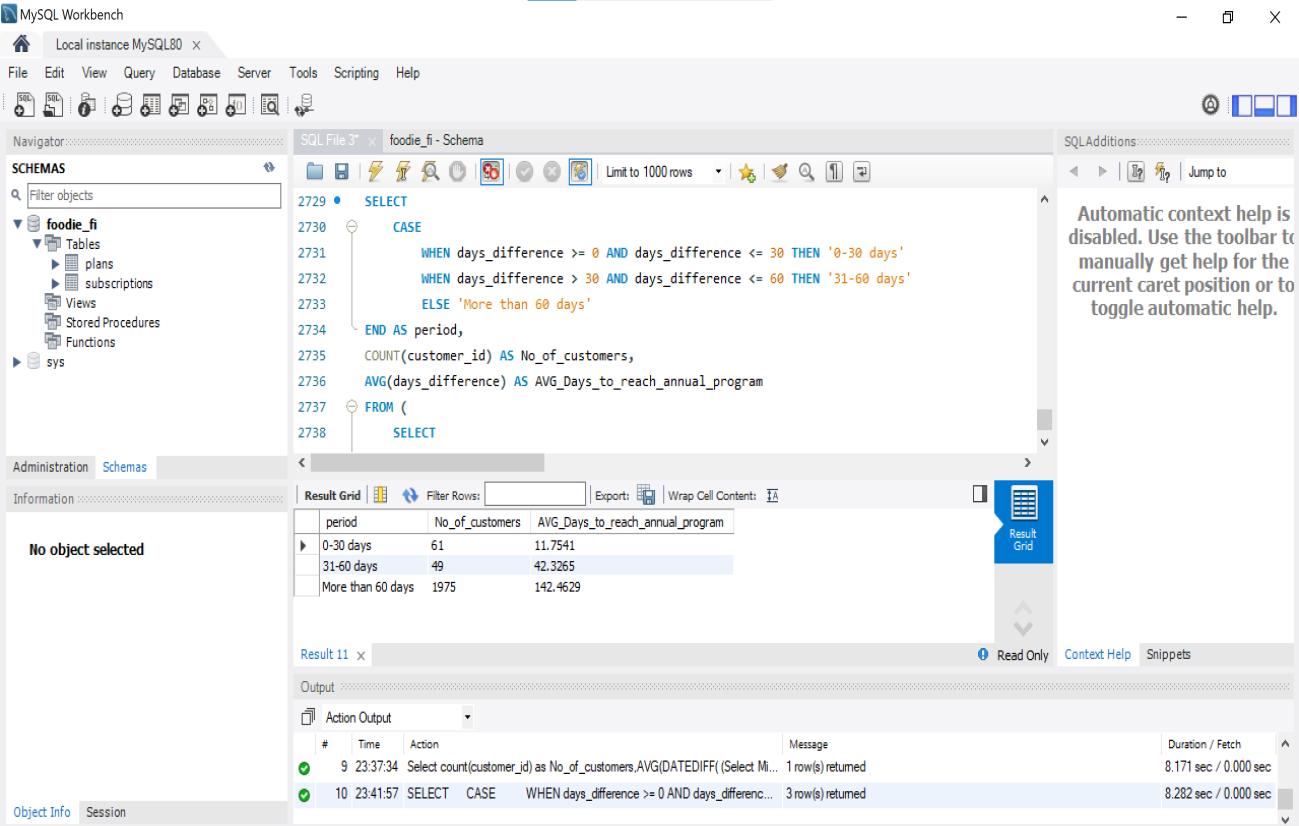
**(SELECT MIN(start\_date) FROM subscriptions AS s3 WHERE s3.customer\_id = s2.customer\_id)) AS**

**days\_difference**

**FROM subscriptions AS s2 WHERE plan\_id ! = 3 AND plan\_id ! = 4) AS differences**

**GROUP BY period**

**ORDER BY period;**



**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);**

