**Case Study – Foodie Fi**

Surya M - Jan 26, 2023



**Introduction**

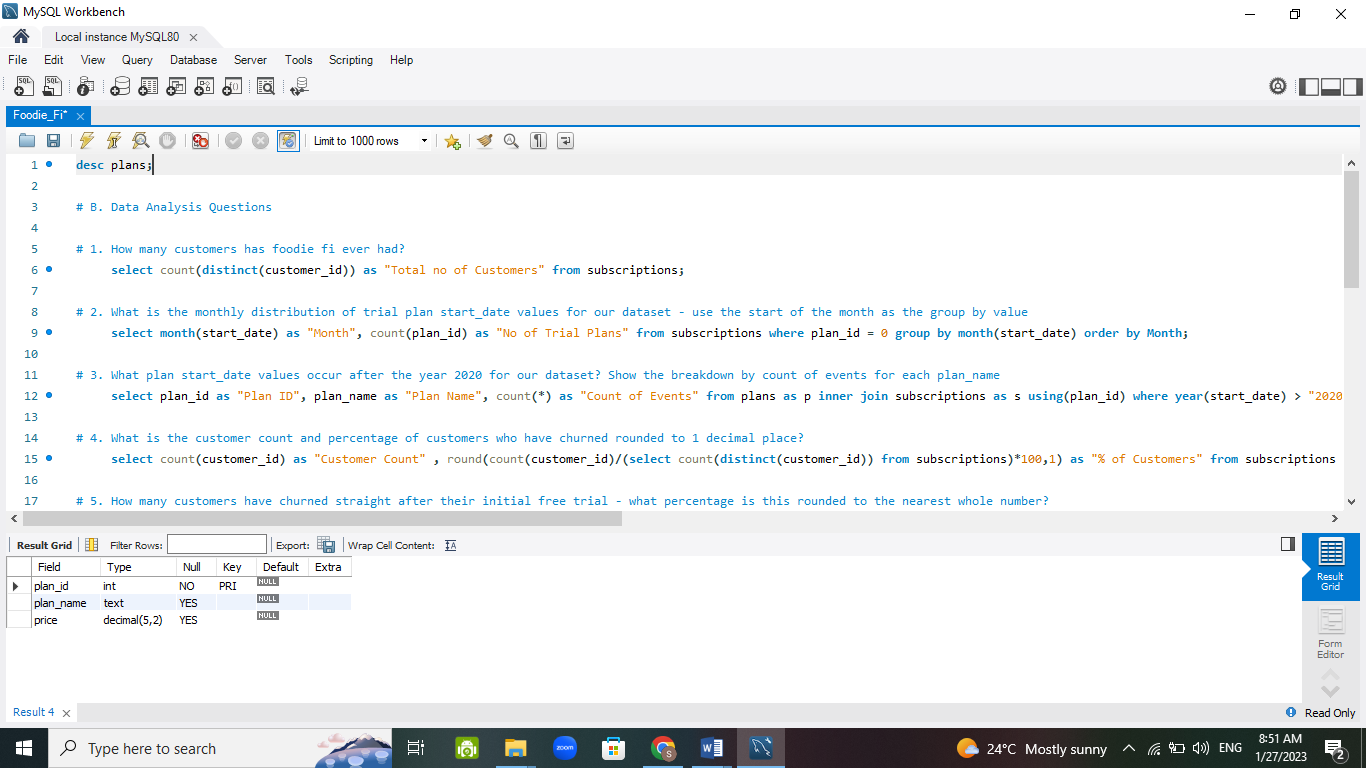
Subscription based businesses are super popular and Danny realised that there was a large gap in the market - he wanted to create a new streaming service that only had food related content - something like Netflix but with only cooking shows!

Danny finds a few smart friends to launch his new startup Foodie-Fi in 2020 and started selling monthly and annual subscriptions, giving their customers unlimited on-demand access to exclusive food videos from around the world!

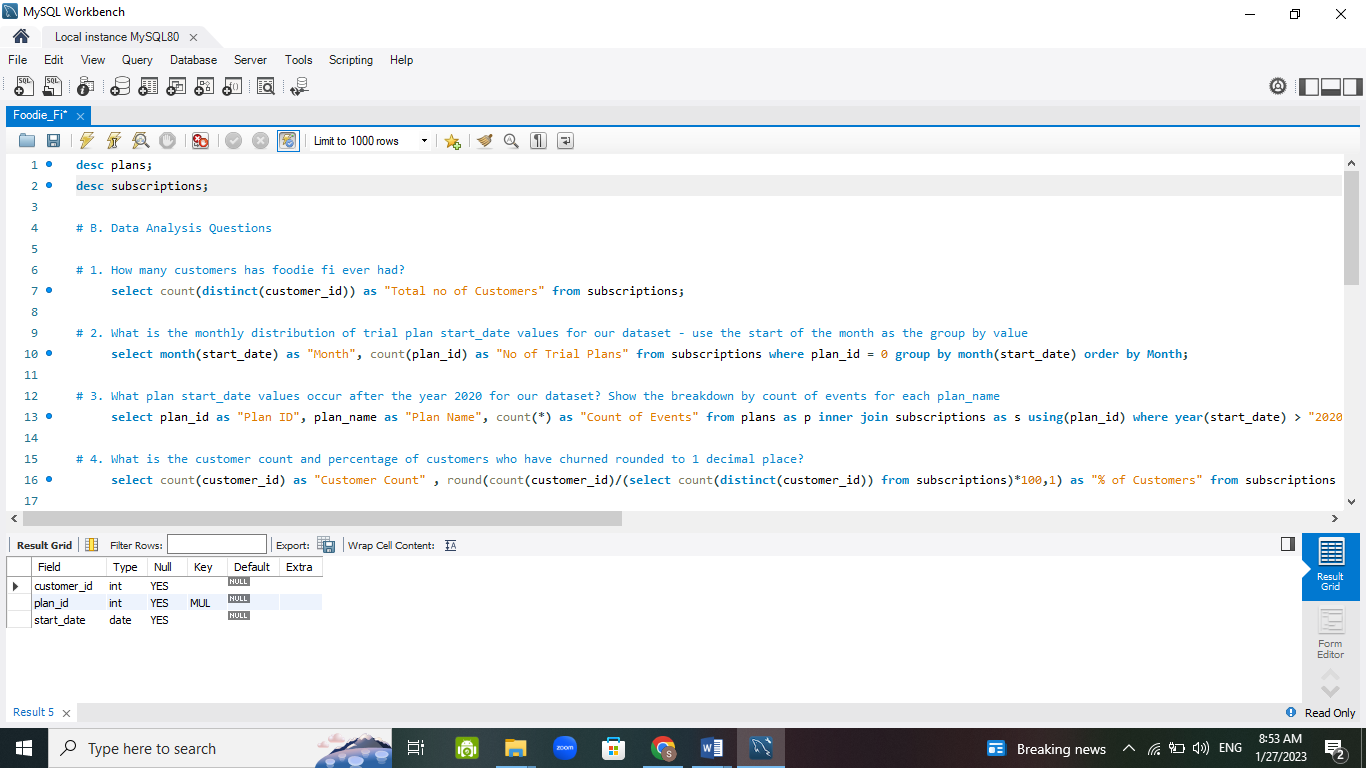
Danny created Foodie-Fi with a data driven mindset and wanted to ensure all future investment decisions and new features were decided using data. This case study focuses on using subscription style digital data to answer important business questions.

**Table Creation**

Plans Table - **CREATE** **TABLE** plans **(**plan\_id **INT**, plan\_name **TEXT**, price **DECIMAL** **(**5,2**))**;



Subscriptions Table - **CREATE TABLE** subscriptions **(**customer\_id **INT**, plan\_id **INT**, start\_date **DATE)**;



**Inserting Values in Tables**

Plans

**INSERT INTO** plans **VALUES**

(0, "trial", 0),

(1, "basic monthly", 9.90),

(2, "pro monthly", 19.90),

(3, "pro annual", 199),

(4, "churn", null);

Subscriptions

**INSERT INTO** subscriptions **VALUES**

(1, 0, '2020-08-01'),

(1, 1, '2020-08-08'),

(2, 0, '2020-09-20'),

(2, 3, '2020-09-27'),

(11, 0, '2020-11-19'),

(11, 4, '2020-11-26'),

(13, 0, '2020-12-15'),

(13, 1, '2020-12-22'),

(13, 2, '2021-03-29'),

(15, 0, '2020-03-17'),

(15, 2, '2020-03-24'),

(15, 4, '2020-04-29'),

(16, 0, '2020-05-31'),

(16, 1, '2020-06-07'),

(16, 3, '2020-10-21'),

(18, 0, '2020-07-06'),

(18, 2, '2020-07-13'),

(19, 0, '2020-06-22'),

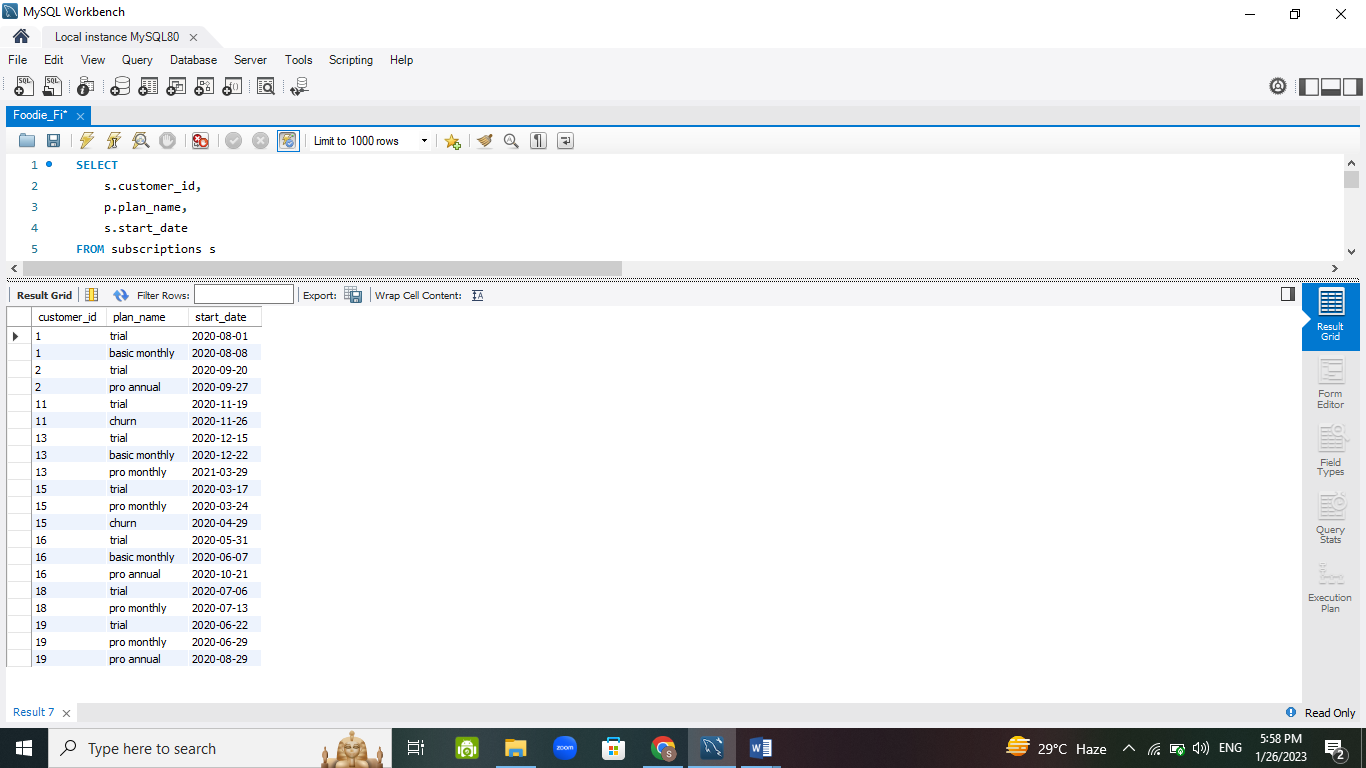
(19, 2, '2020-06-29'),

(19, 3, '2020-08-29');

**Case** **Study** **Questions**

1. **Customer’s** **Journey**

**SELECT** s.customer\_id, p.plan\_name, s.start\_date **FROM** subscriptions **AS** s **INNER JOIN** plans **AS** p using (plan\_id) **ORDER BY** s.customer\_id;



* Customer ID 1’s Journey

Customer 1's journey starts with the trial on 2020-08-01, and when the trial ends they upgrade to the basic monthly plan on 2020-08-08.

* Customer ID 2’s Journey

Customer 2's journey starts with the trial on 2020-09-20, and when the trial ends they upgrade to the pro annual plan on 2020-09-27.

* Customer ID 11’s Journey

Customer 11's journey starts with the trial on 2020-11-19, and when the trial ends they churned the subscriptions on 2020-11-26.

* Customer ID 13’s Journey

Customer 13's journey starts with the trial on 2020-12-15, and when the trial ends they upgrade to the basic monthly plan on 2020-12-22. After 3 months, upgraded to pro monthly plan on 2021-03-29.

* Customer ID 15’s Journey

Customer 15's journey starts with the trial on 2020-03-17, and when the trial ends they upgrade to the pro monthly plan on 2020-03-24. After a month they churn the subscriptions on 2020-04-29.

* Customer ID 16 Journey

Customer 16's journey starts with the trial on 2020-05-31, and when the trial ends they upgrade to the basic monthly plan on 2020-06-07. After 4 months, they again upgraded to pro annual plan on 2021-10-21.

* Customer ID 18 Journey

Customer 18's journey starts with the trial on 2020-07-06, and when the trial ends they upgrade to the pro annual plan on 2020-10-21.

* Customer ID 19 Journey

Customer 19's journey starts with the trial on 2020-06-22, and when the trial ends they upgrade to the pro monthly plan on 2020-06-29. After 2 months, they again upgraded to the pro annual plan on 2020-08-29.

1. **Data Analysis Questions**

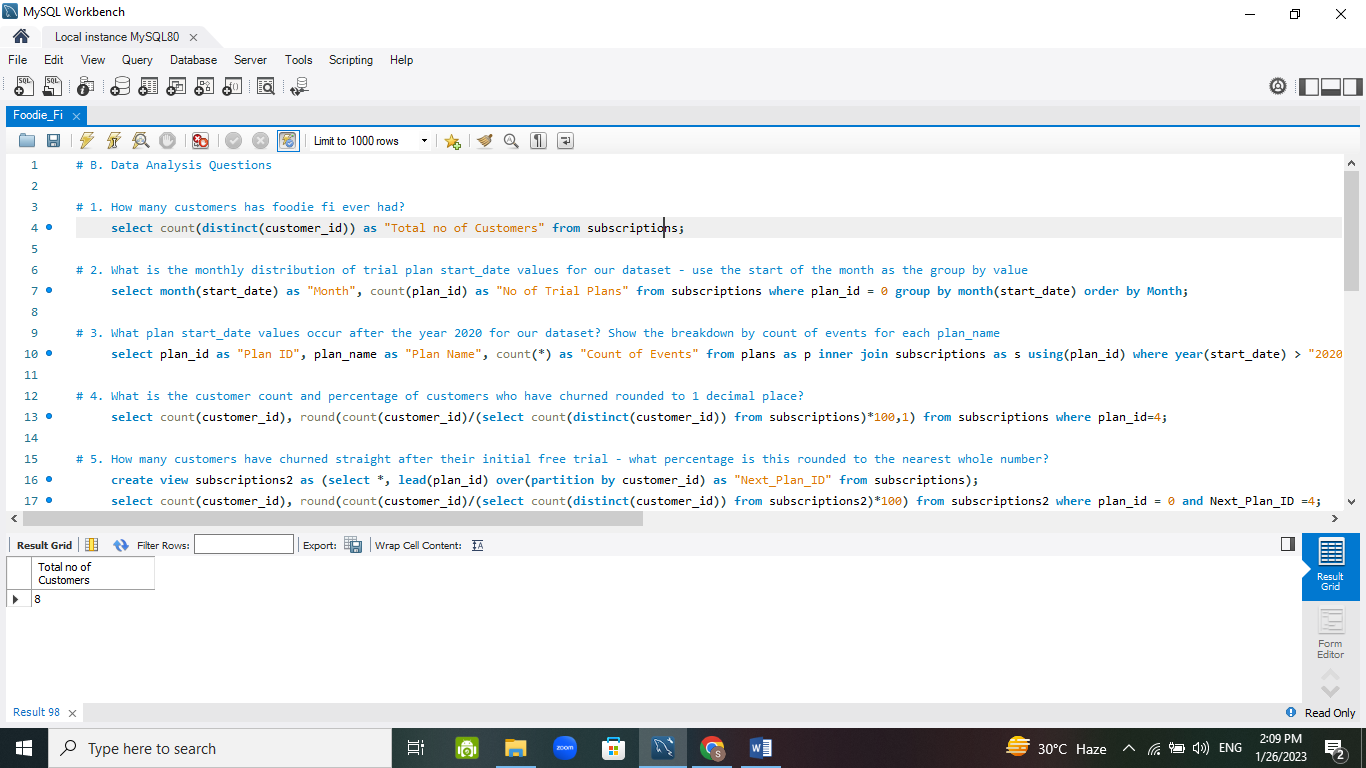
* How many customers has foodie fi ever had?

**SELECT**

count(distinct( customer\_id )) **AS** "Total no of Customers"

**FROM** subscriptions;

We can use COUNT and DISTINCT to return different values of customer\_id.



* 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** "Month",

count(plan\_id ) **AS** "No of Trial Plans"

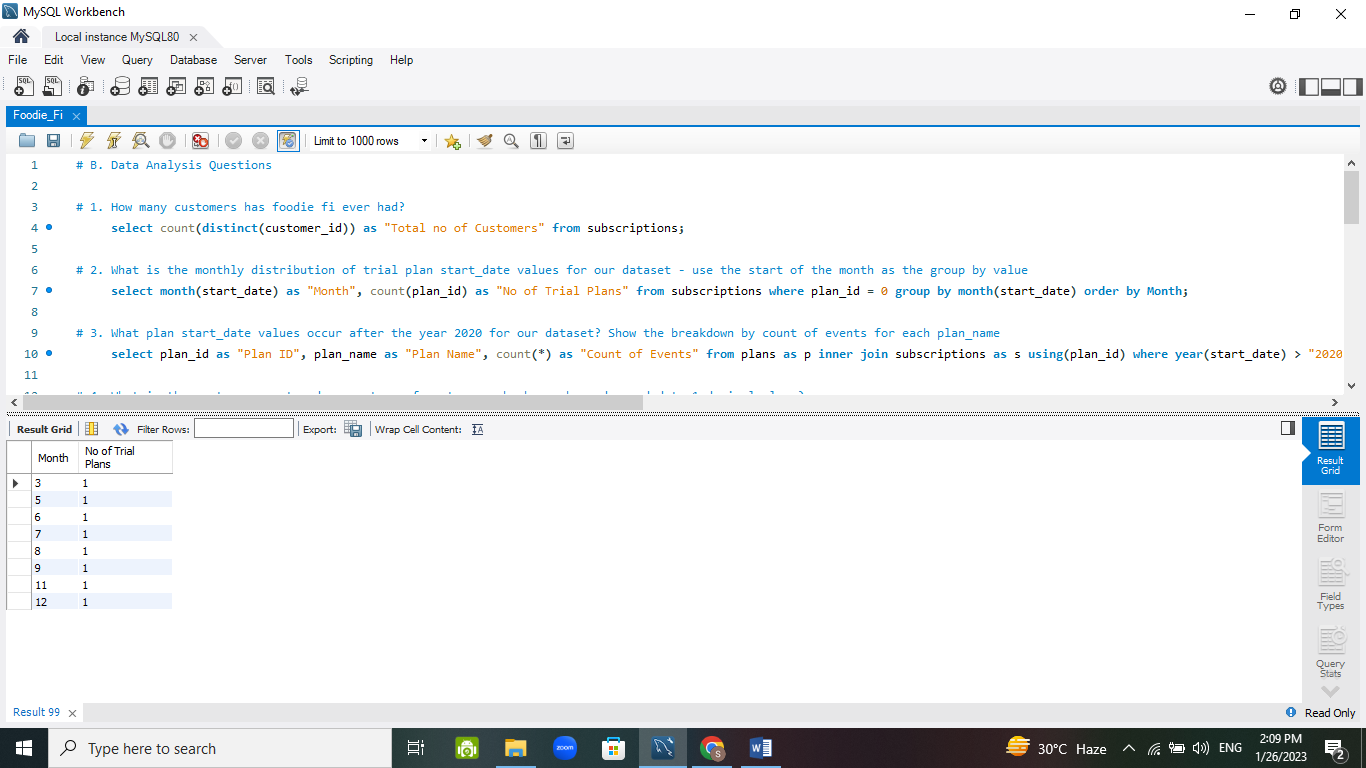
**FROM** subscriptions

**WHERE** plan\_id = 0

**GROUP BY** month(start\_date )

**ORDER BY** Month;

We can use MONTH statement to extract the month of start\_date and COUNT(plan\_id) for finding the monthly distribution.



* 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\_id **AS** "Plan ID",

plan\_name **AS** "Plan Name",

count (\*) **AS** "Count of Events"

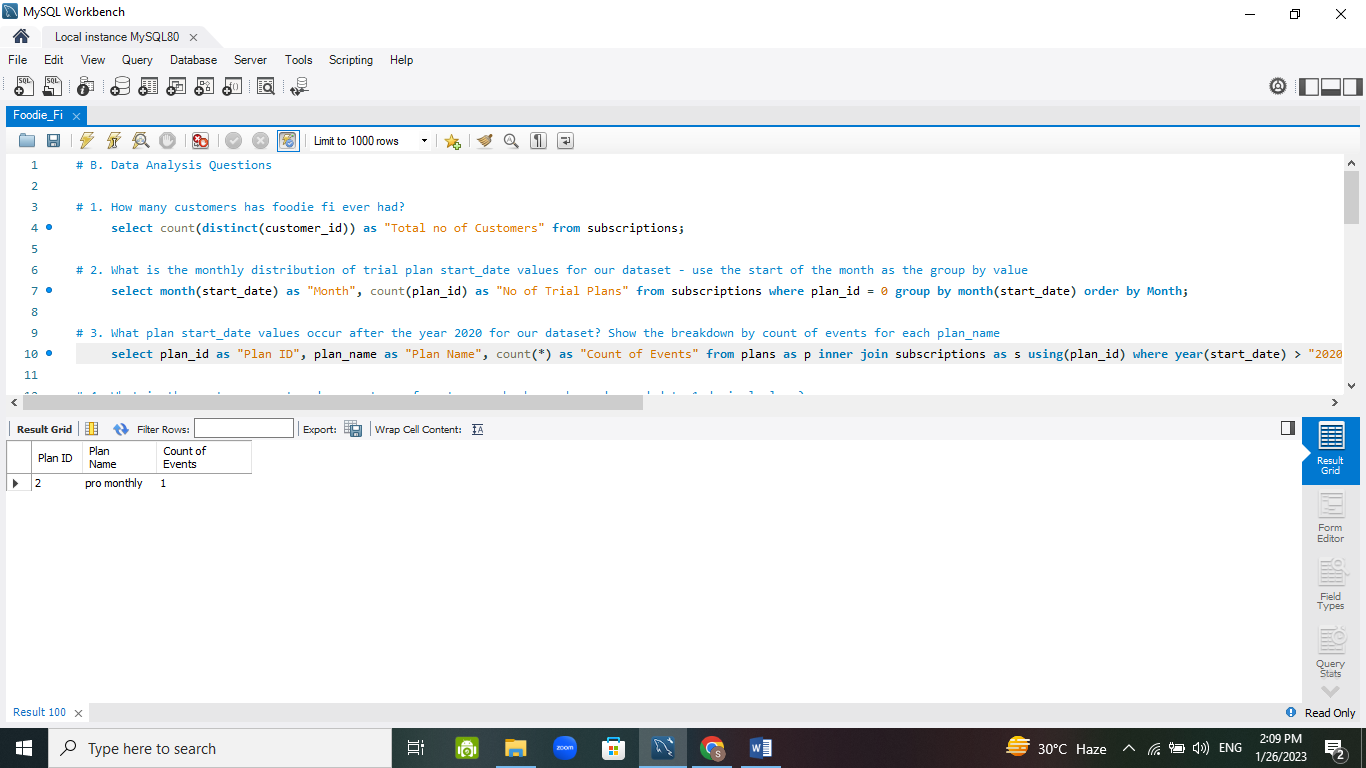
**FROM** plans **AS** p

**INNER JOIN** subscriptions **AS** s using( plan\_id )

**WHERE** year( start\_date ) > "2020"

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

We have to join plans and subscriptions on plan\_id and condition is year of start\_date should be greater than 2020.



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

**SELECT**

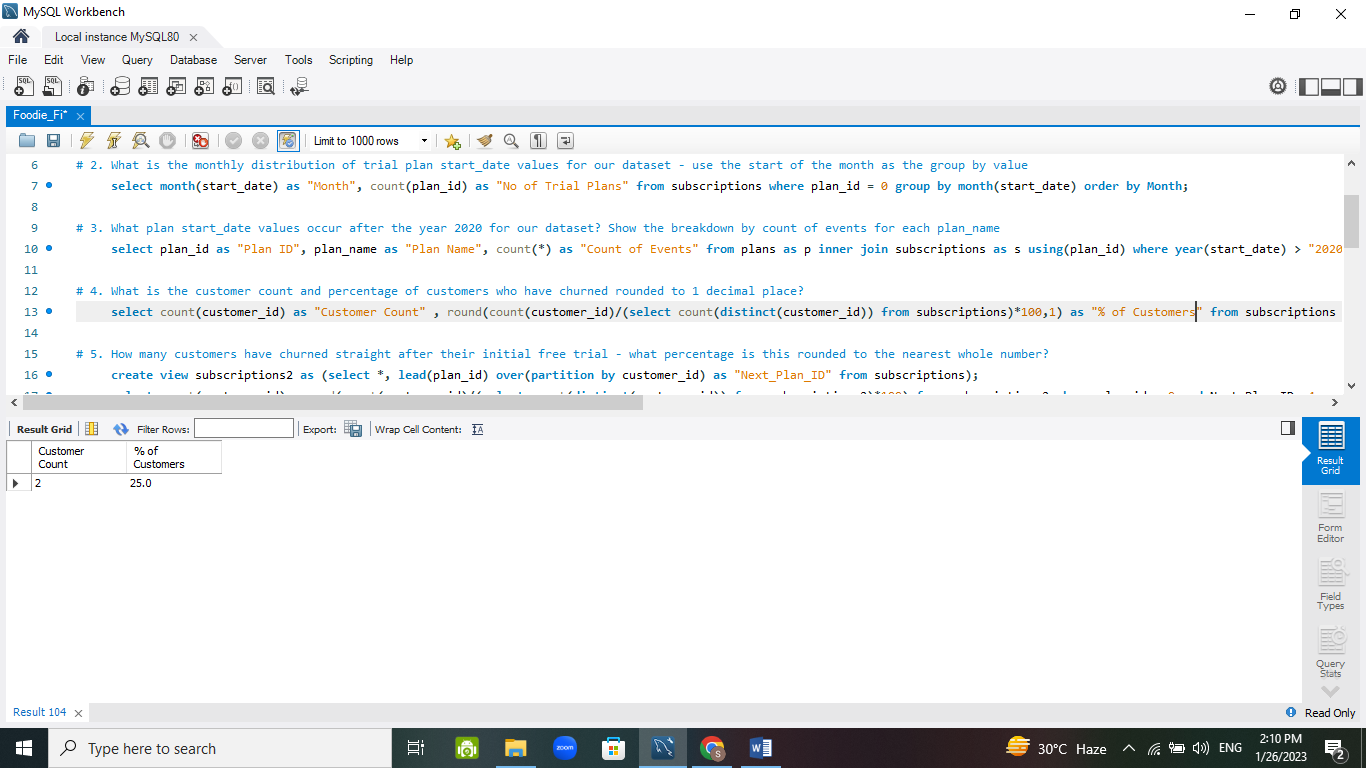
count( customer\_id ) **AS** "Customer Count" ,

round(count( customer\_id ) **/** ( **SELECT** count(distinct( customer\_id )) **FROM** subscriptions) \*100,1 ) **AS** "% of Customers"

**FROM** subscriptions

**WHERE** plan\_id=4;

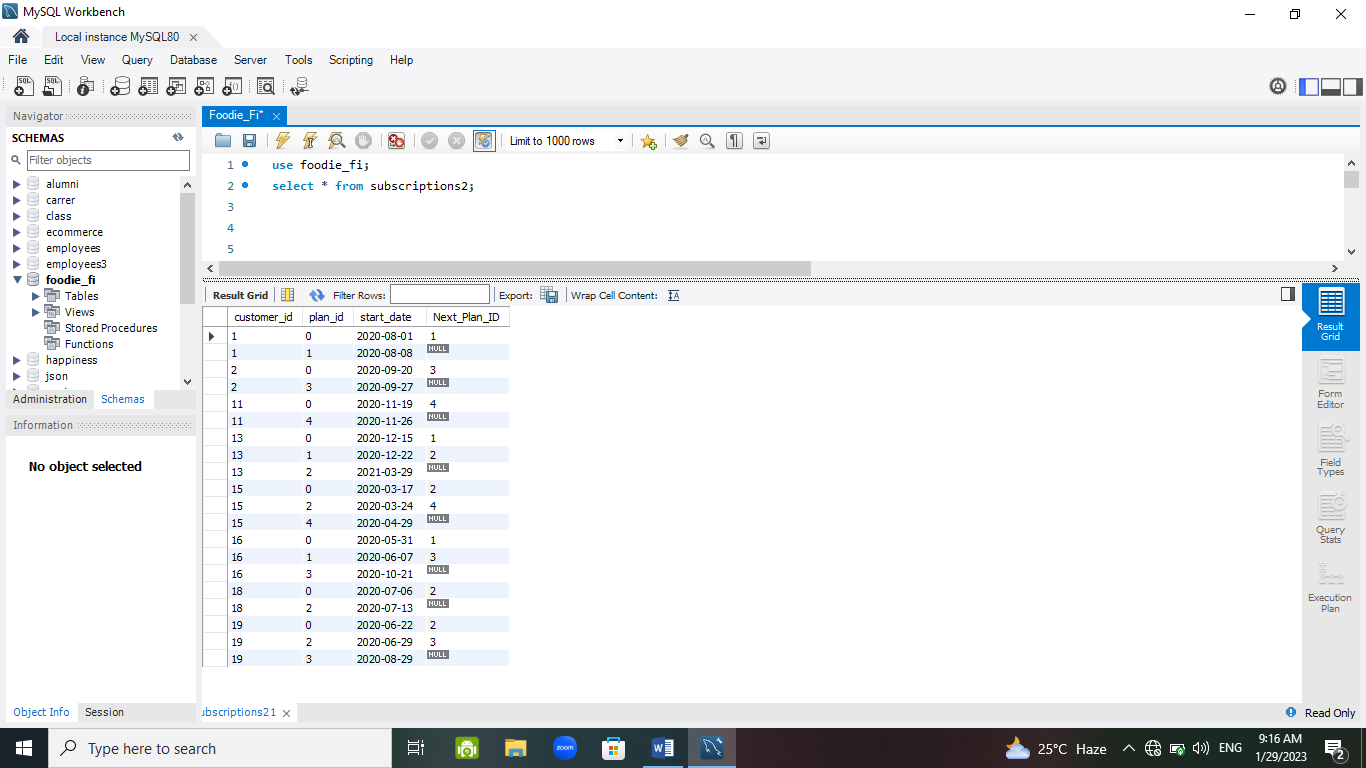
We have to count the customer ID and % of Customer ID Churned from subscriptions and condition is plan\_id = 4.



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

**CREATE** view subscriptions2

**AS** ( **SELECT** \*, lead( plan\_id ) over ( **PARTITION BY** customer\_id ) **AS** "Next\_Plan\_ID" **FROM** subscriptions );



**SELECT**

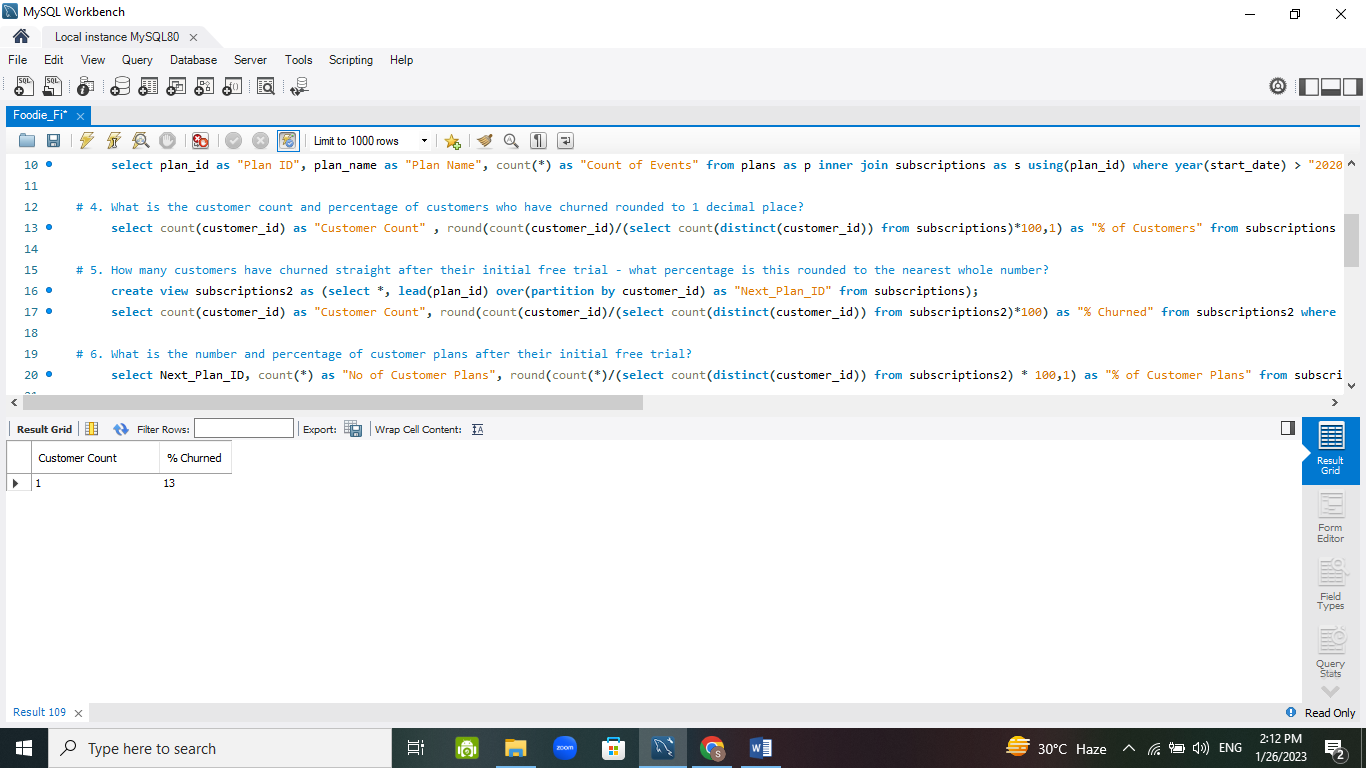
count( customer\_id ) **AS** "Customer Count",

round(count( customer\_id ) **/** ( **SELECT** count(distinct( customer\_id )) **FROM** subscriptions2 ) \*100 ) **AS** "% Churned"

**FROM** subscriptions2

**WHERE** plan\_id = 0 and Next\_Plan\_ID =4;

We are creating a virtual table subscritpions2 where we are adding another column Next\_Plan\_ID which is the lead of plan\_id and then we are finding count of customer ID and % of customer ID churned with the condition plan\_id = 0 and Next\_Plan\_ID = 4.



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

**SELECT**

Next\_Plan\_ID,

count (\*) **AS** "No of Customer Plans",

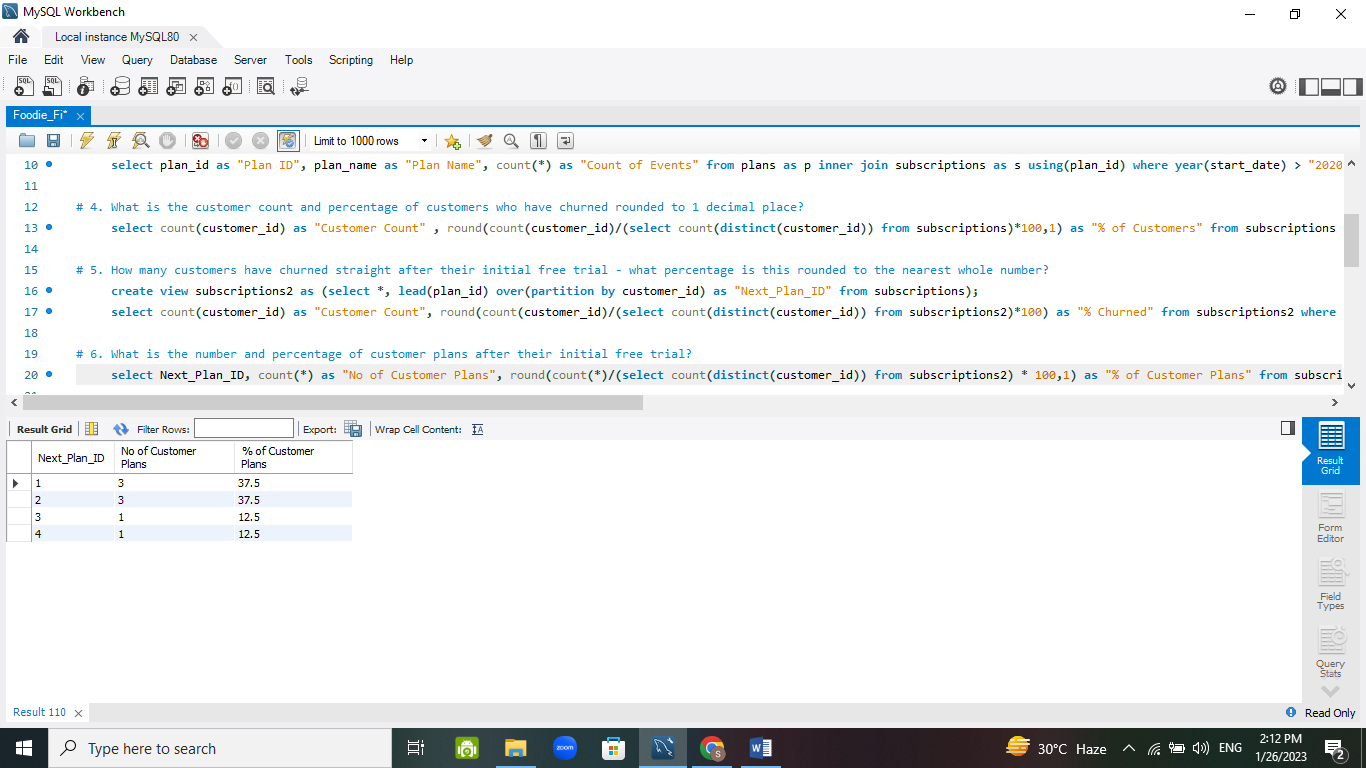
round (count (\*) **/** ( **SELECT** count(distinct( customer\_id )) **FROM** subscriptions2 ) \* 100,1 ) **AS** "% of Customer Plans"

**FROM** subscriptions2

**WHERE** plan\_id = 0 and (Next\_Plan\_ID = 1 or Next\_Plan\_ID = 2 or Next\_Plan\_ID = 3 or Next\_Plan\_ID = 4)

**GROUP** **BY** Next\_Plan\_ID;

We are finding the count of customers and % of customers plans from subscriptions2 where plan\_id = 0 (i.e. free trial) but Next\_Plan\_ID can be 1,2,3 or 4 by grouping on the basis of Next\_Plan\_ID.



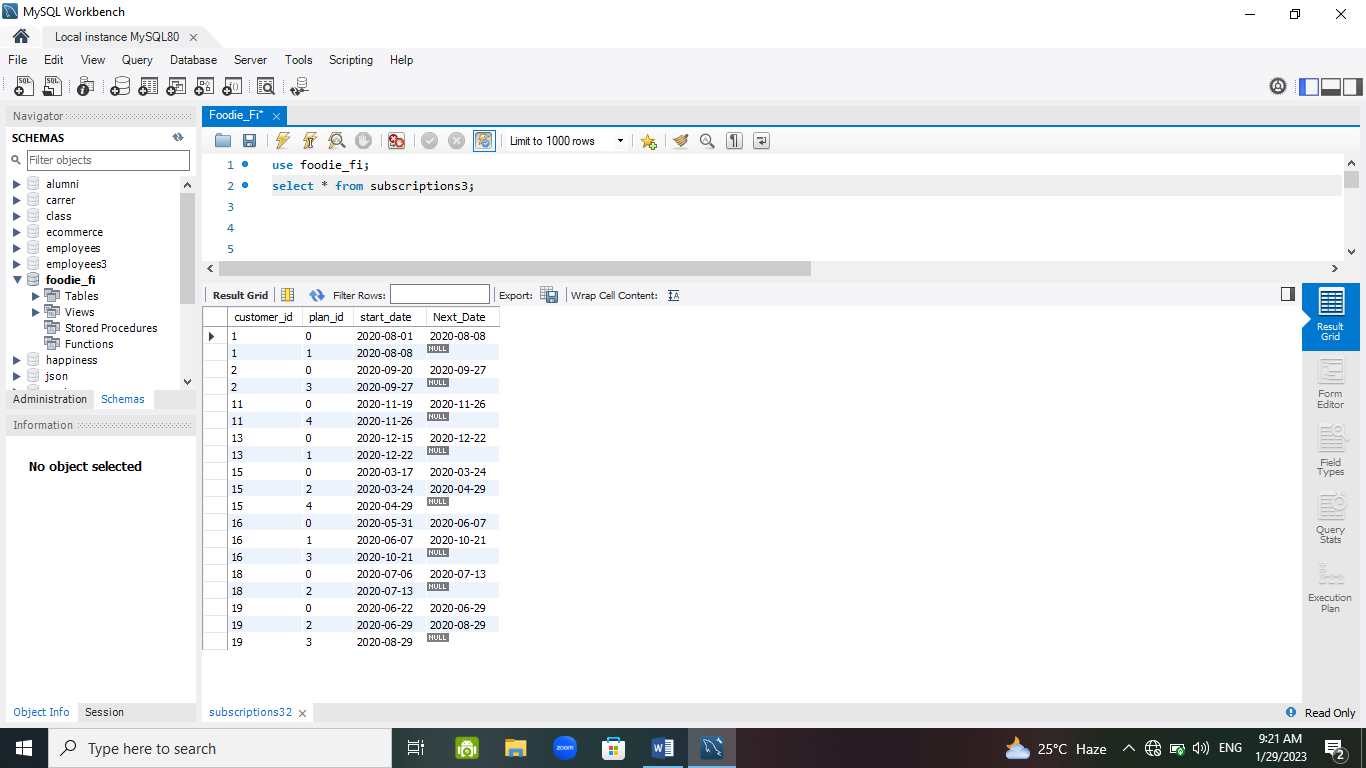
* What is the customer count and percentage breakdown of all 5 plan\_name values at 2020-12-31?

**CREATE** view subscriptions3

**AS** ( **SELECT** \*, lead( start\_date ) over ( **PARTITION BY** customer\_id ) **AS** "Next\_Date"

**FROM** subscriptions

**WHERE** start\_date <= "2020-12-31" );



**SELECT**

plan\_id **AS** "Plan ID",

count(distinct( customer\_id )) **AS** "Total Customers",

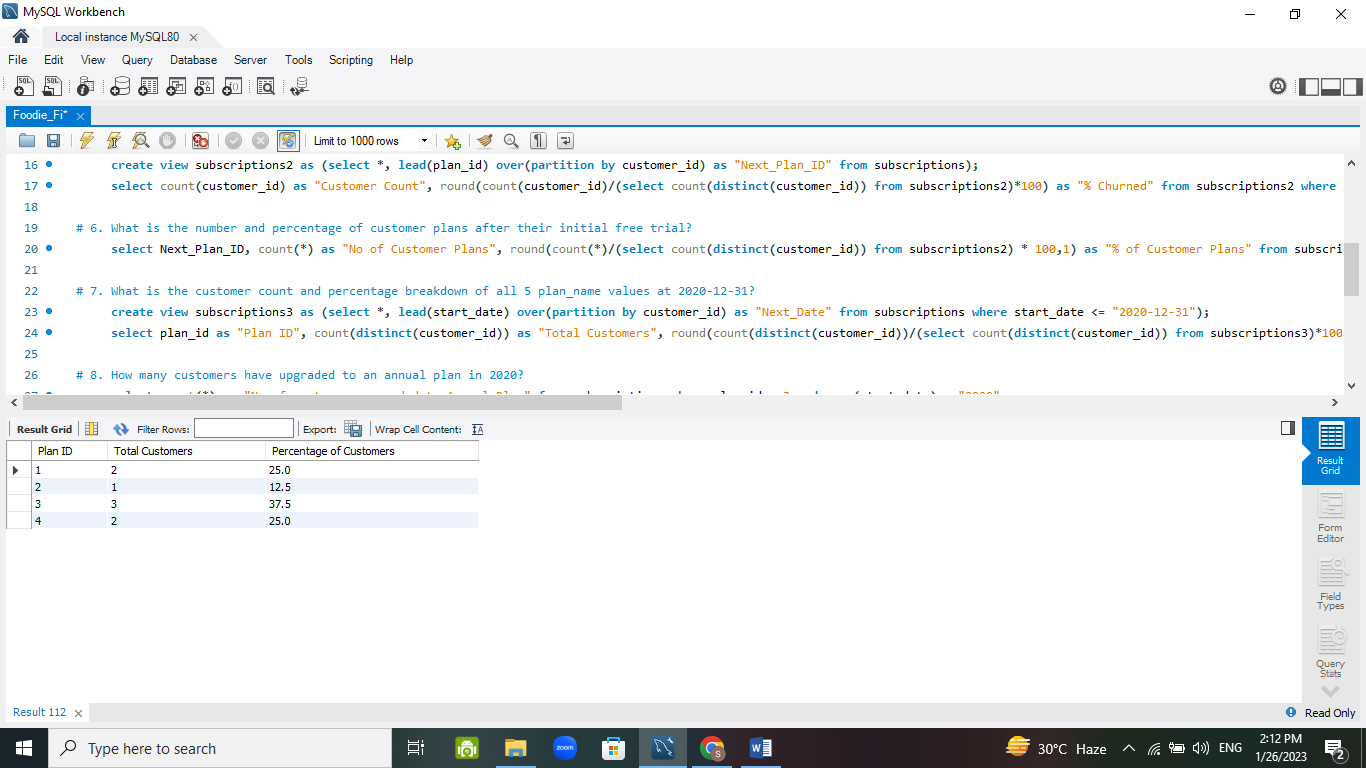
round(count(distinct( customer\_id )) **/** ( **SELECT** count(distinct( customer\_id )) **FROM** subscriptions3 ) \*100,1 ) **AS** "Percentage of Customers"

**FROM** subscriptions3

**WHERE** ( start\_date < "2020-12-31" and Next\_Date > "2020-12-31" and Next\_Date is not null ) or ( Next\_Date is null and start\_date < "2020-12-31" )

**GROUP BY** plan\_id;

We are creating a virtual table subscriptions3 where we are adding another column Next\_Date which is the lead of start\_date and then finding the customer count and percentage breakdown where condition is start date less than 2020-12-31 and Next Date greater than 2020-12-31 and Next Date is not null.



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

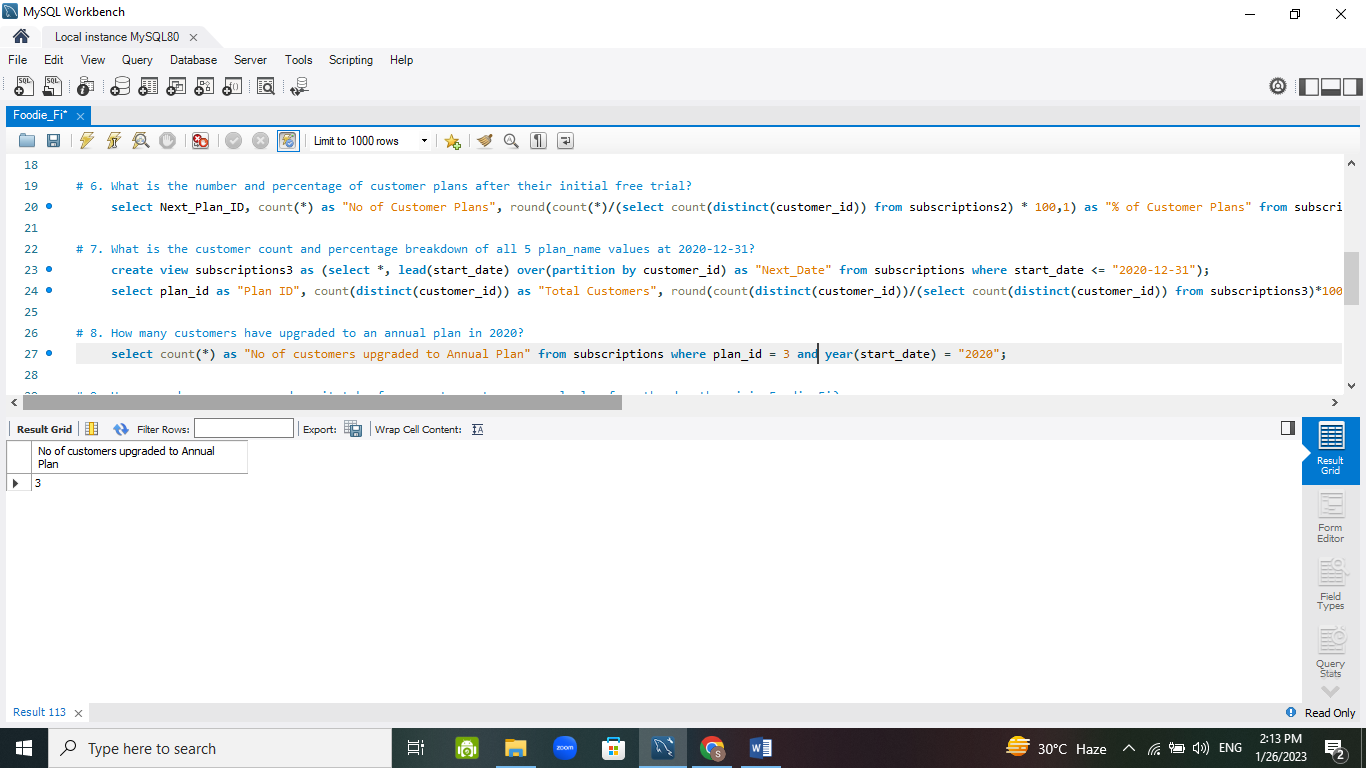
**SELECT**

count (\*) **AS** "No of customers upgraded to Annual Plan"

**FROM** subscriptions

**WHERE** plan\_id = 3 and year(start\_date) = "2020";

We are selecting count (\*) for customer count who have upgraded to annual plan where plan\_id = 3 and in the year 2020.



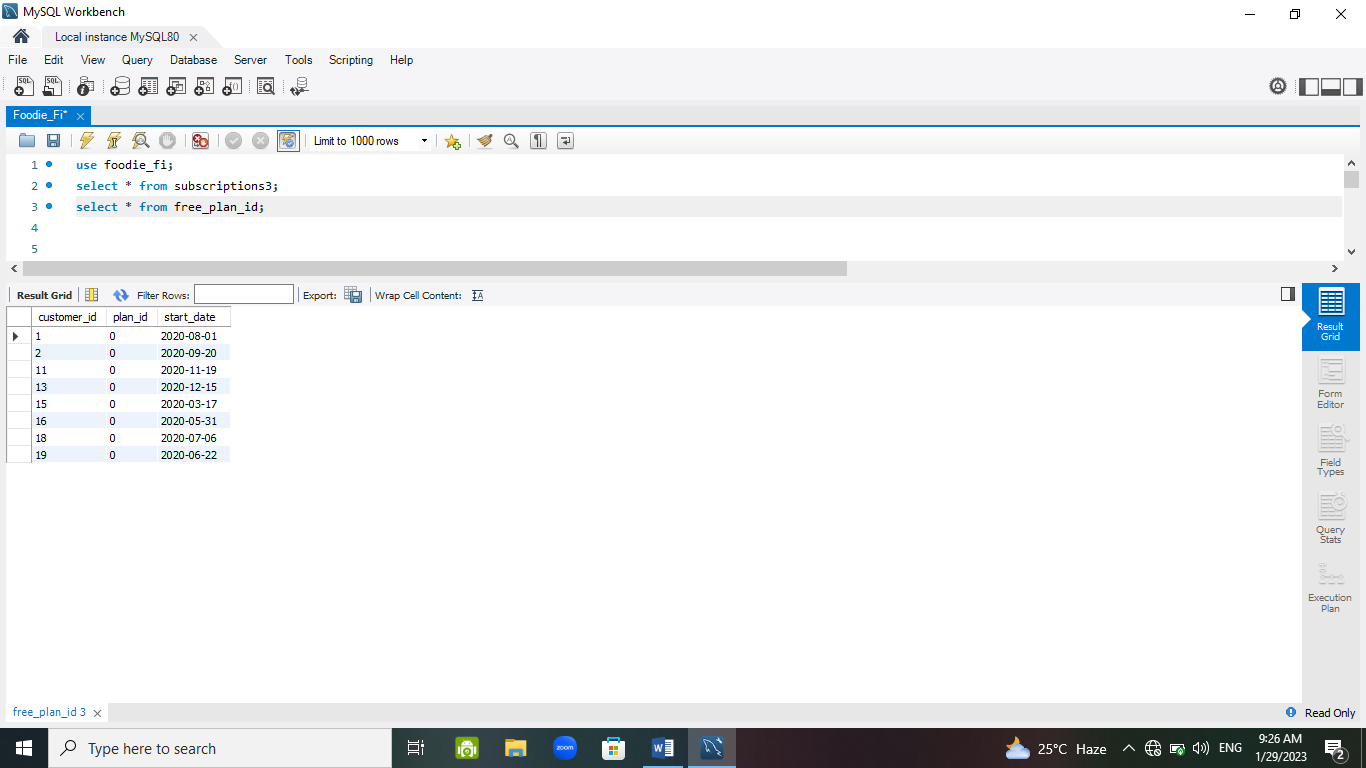
* How many days on average does it take for a customer to an annual plan FROM the day they join Foodie-Fi?

**CREATE** view free\_plan\_id

**AS** ( **SELECT** customer\_id, plan\_id, start\_date

**FROM** subscriptions

**WHERE** plan\_id = 0 );

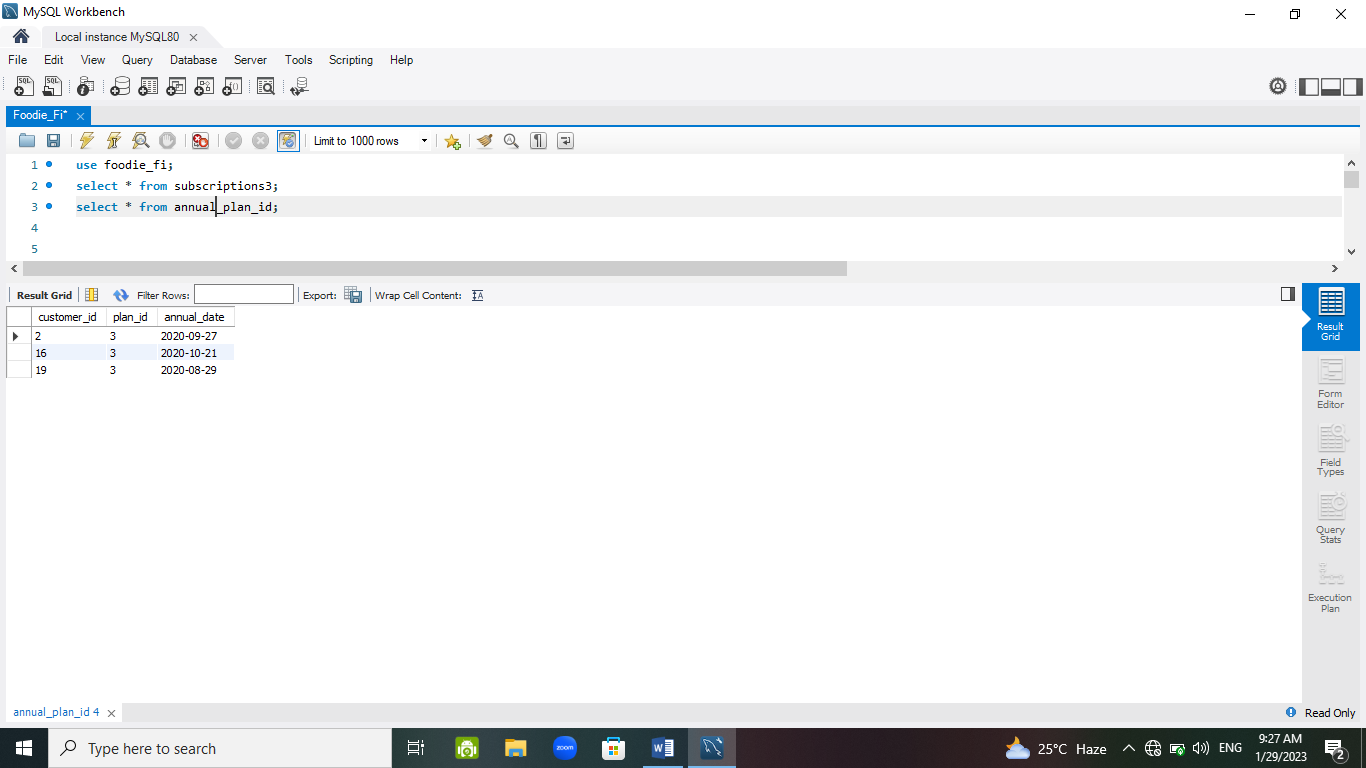


**CREATE** view annual\_plan\_id

**AS** ( **SELECT** customer\_id, plan\_id, start\_date **AS** "annual\_date"

**FROM** subscriptions

**WHERE** plan\_id = 3 );



**SELECT**

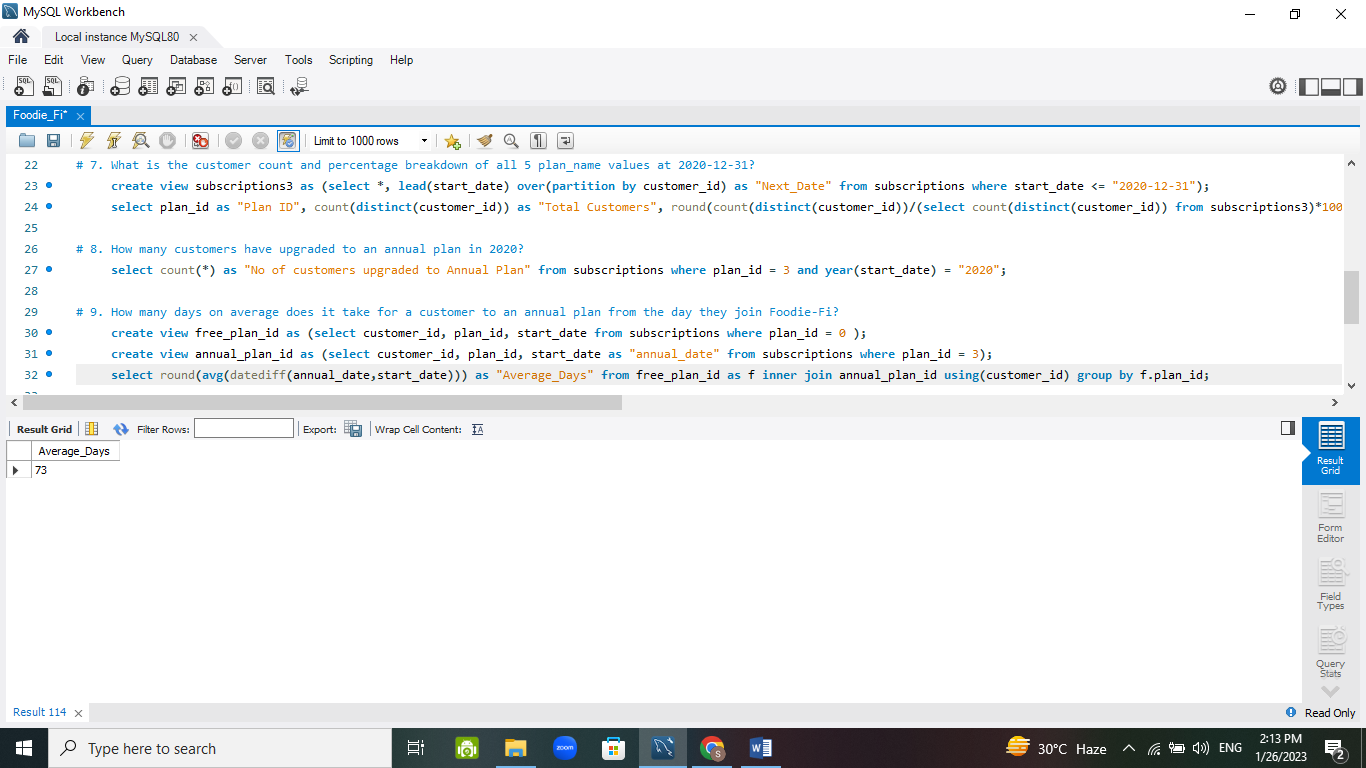
round(avg(datediff( annual\_date,start\_date ))) **AS** "Average\_Days"

**FROM** free\_plan\_id

**INNER JOIN** annual\_plan\_id using( customer\_id )

**GROUP BY** f.plan\_id;

We are creating 2 virtual tables free\_plan\_id and annual\_plan\_id where we are defining a condition i.e. plan\_id = 0 and plan\_id = 3 respectively and then we are joining the two tables and finding the average days.



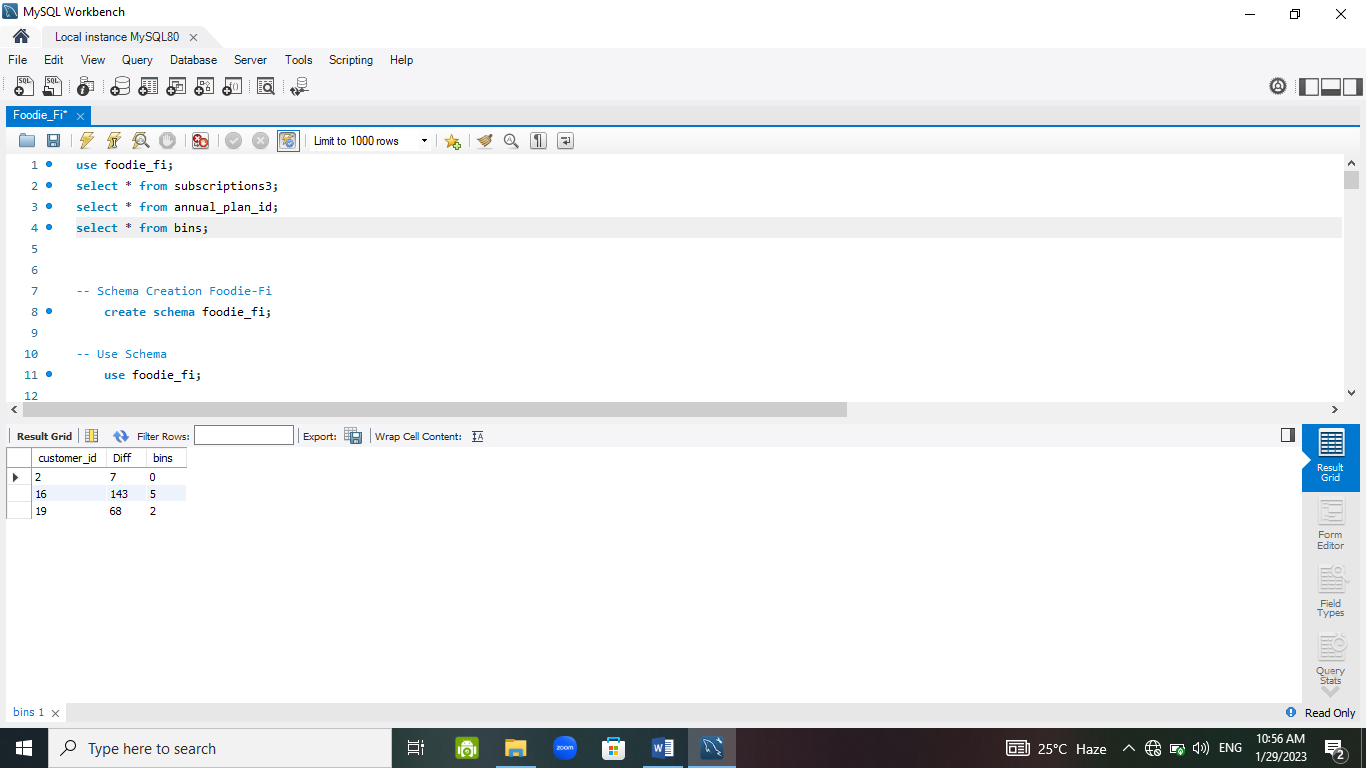
* Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc.)

**CREATE** view bins

**AS** ( **SELECT** customer\_id, datediff(annual\_date,start\_date) **AS** "Diff", round(datediff( annual\_date,start\_date )/30) **AS** "bins"

**FROM** free\_plan\_id **AS** f

**JOIN** annual\_plan\_id using( customer\_id ));



**SELECT**

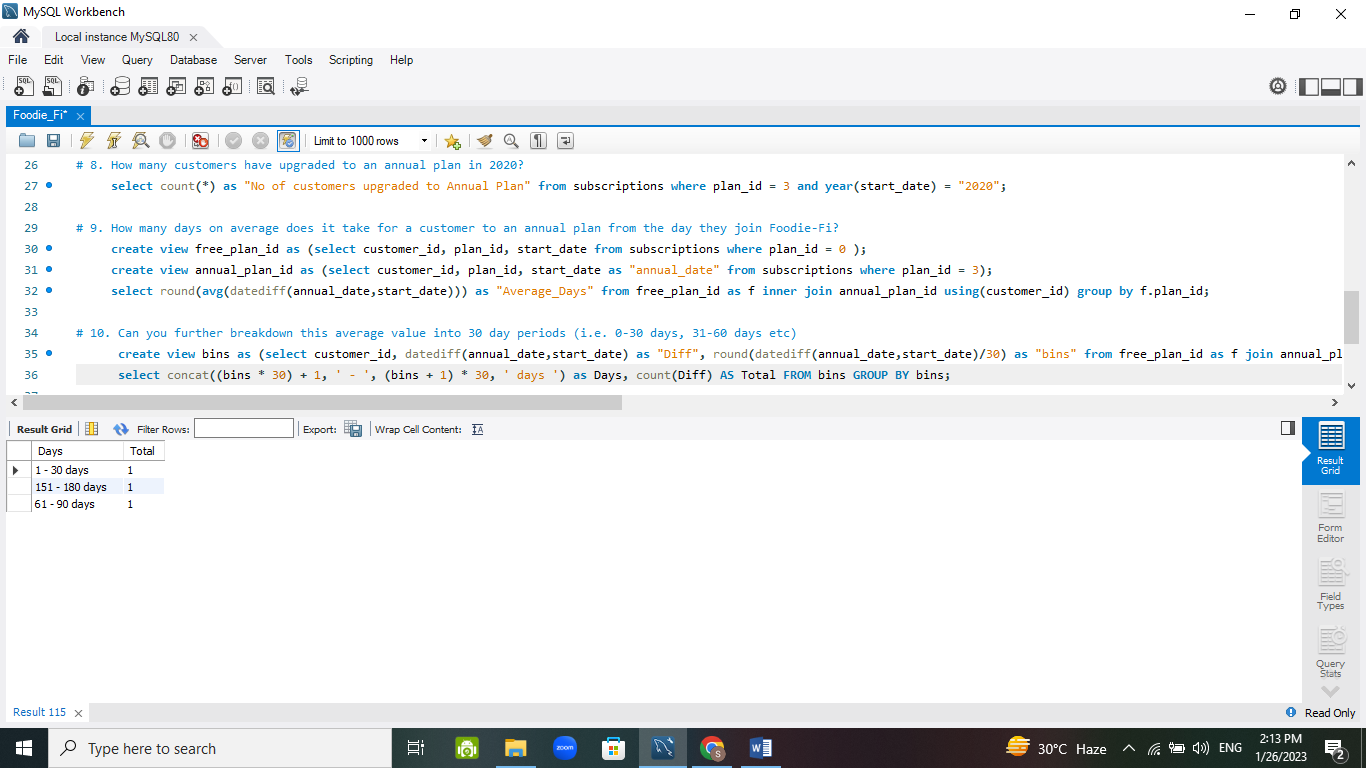
concat (( bins \* 30 ) + 1, ' - ', ( bins + 1 ) \* 30, ' days ') **AS** Days,

count( Diff ) **AS** Total

**FROM** bins

**GROUP BY** bins;

We are creating a virtual table bins where we are finding the days between annual date and start date by joining the two virtual tables free\_plan\_id and annual\_plan\_id on customer\_id and then concat the days interval and count the days from bins table by grouping with bins column.



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

**SELECT**

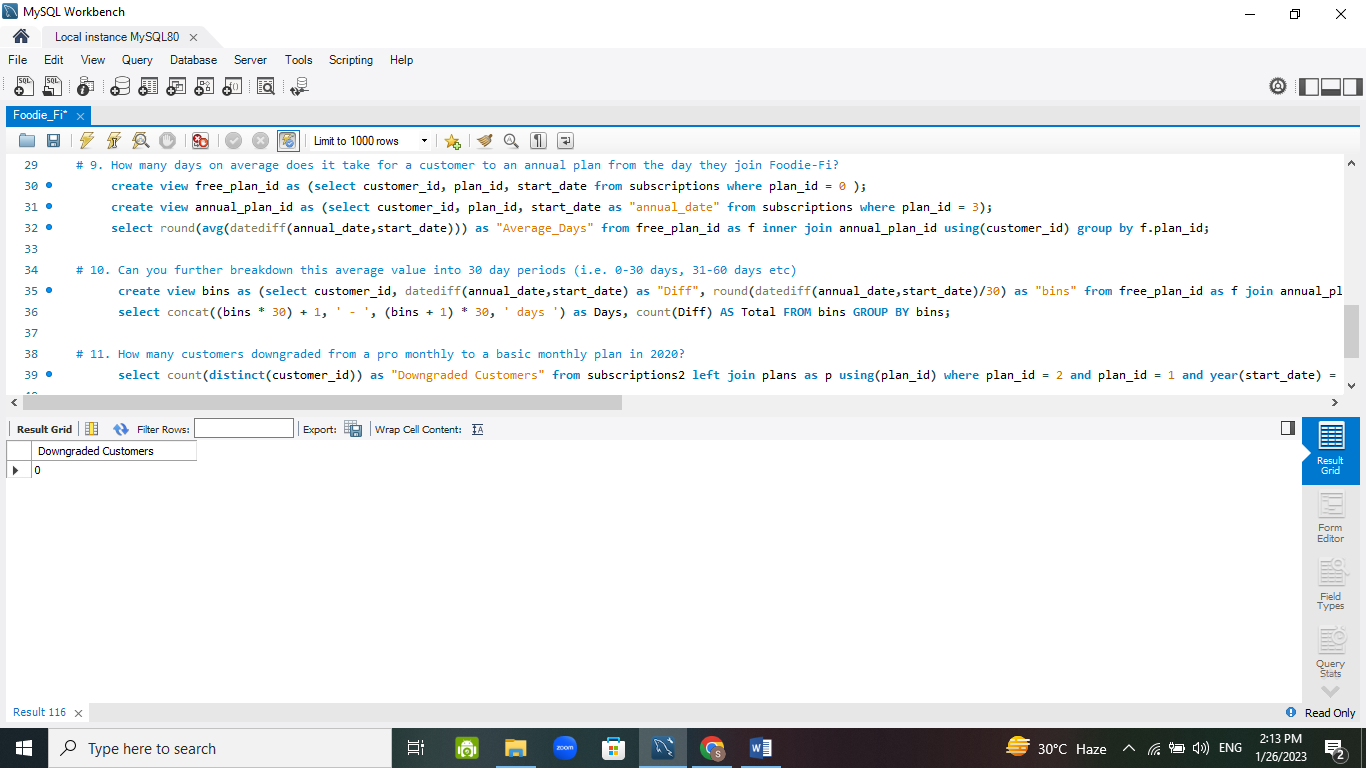
count(distinct(customer\_id )) **AS** "Downgraded Customers"

**FROM** subscriptions2

**LEFT JOIN** plans using(plan\_id )

**WHERE** plan\_id = 2 and plan\_id = 1 and year(start\_date) = "2020";

We are left joining subscriptions2 and plans table using plan\_id and the condition is plan\_id = 2 and plan\_id = 1 and the year of start date is 2020 for finding the customers downgraded.



1. **Challenge Payment Questions**

**Customer ID 1:**

**CREATE** view customer\_id1 **AS** (

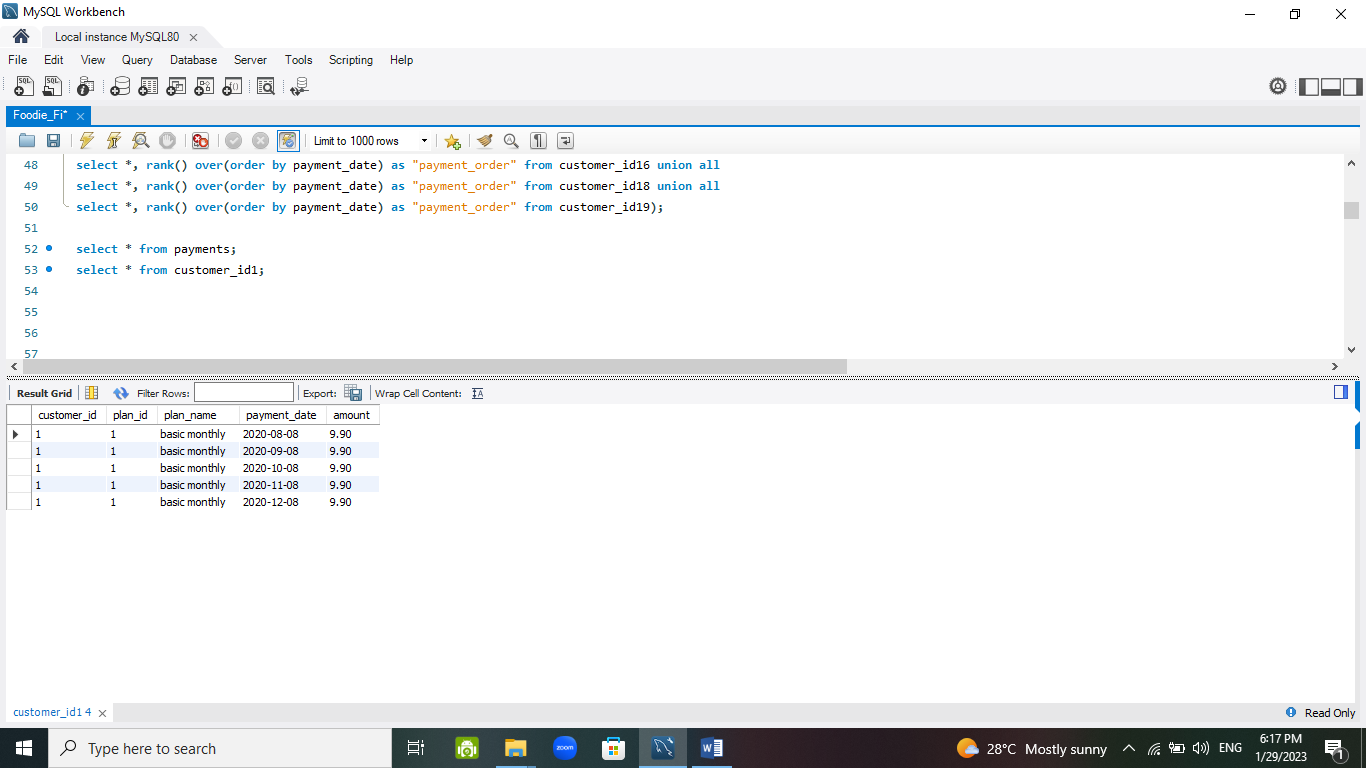
(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id =1) **UNION** **ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 1 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 1) **UNION** **ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 2 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 1) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 3 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 1) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 4 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 1));

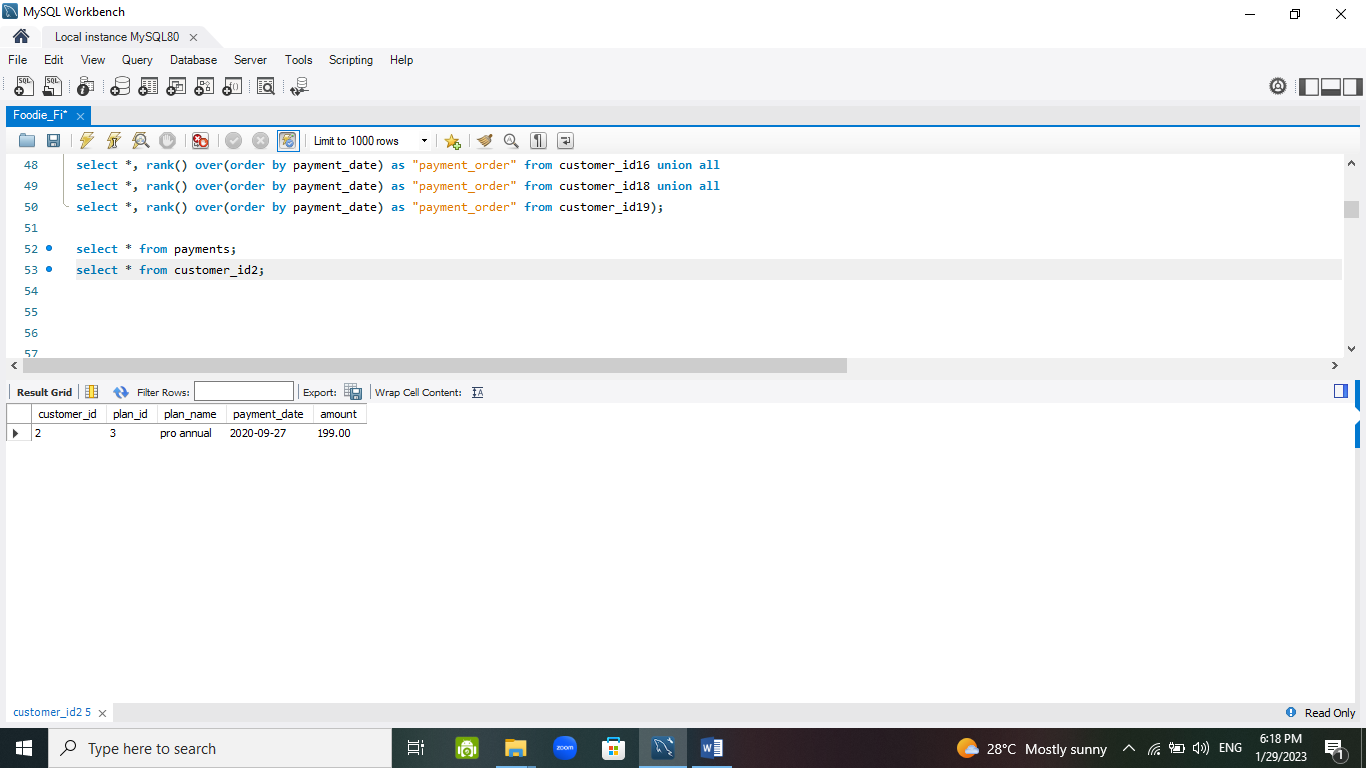


------------------------------------------------------------------------------------------------------------------------------------------------

**Customer ID 2:**

**CREATE** view customer\_id2 **AS** (

(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 2));

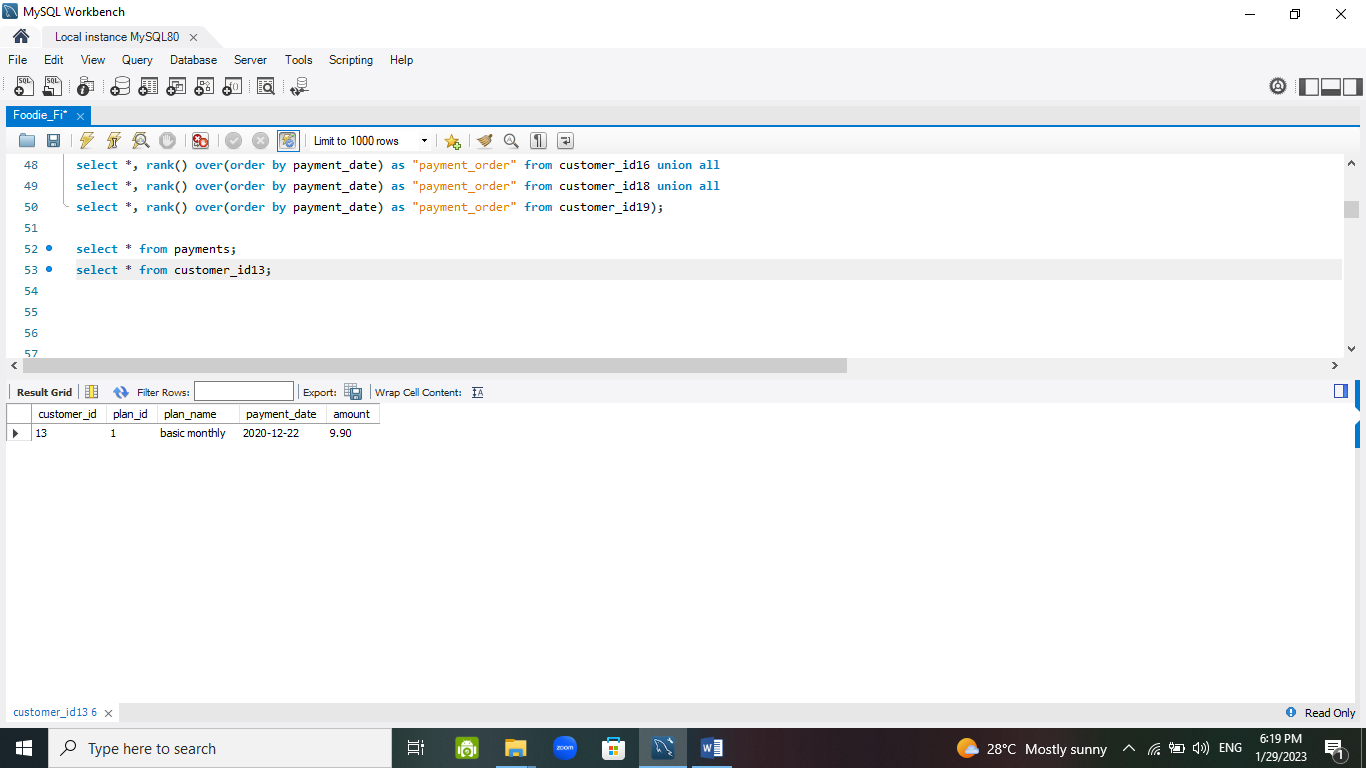


------------------------------------------------------------------------------------------------------------------------------------------------

**Customer ID 13:**

**CREATE** view customer\_id13 **AS** (

(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using (plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 13 **LIMIT** 1));



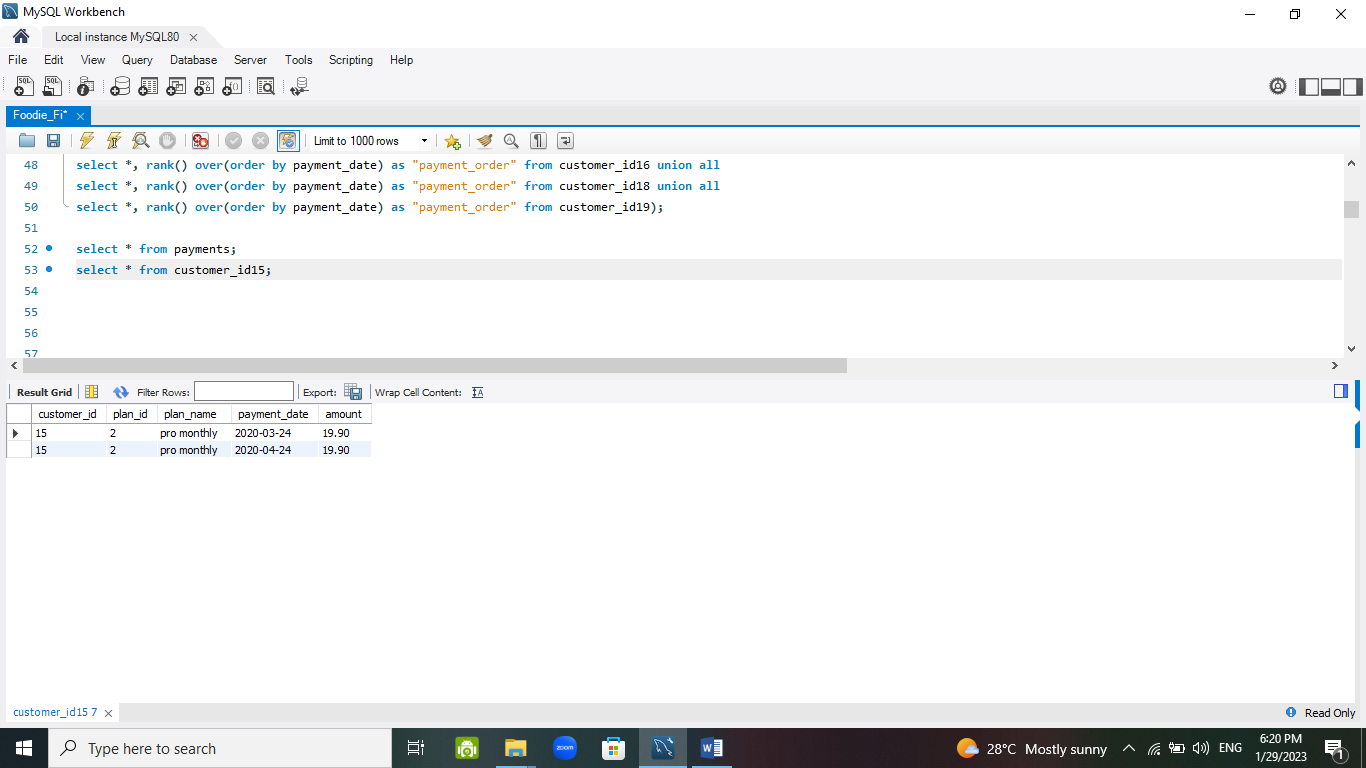
------------------------------------------------------------------------------------------------------------------------------------------------

**Customer ID 15:**

**CREATE** view customer\_id15 **AS** (

(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 15) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 1 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 15));



------------------------------------------------------------------------------------------------------------------------------------------------

**Customer ID 16:**

**CREATE** view customer\_id16 **AS** (

(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 16 **LIMIT** 1) **UNION ALL**

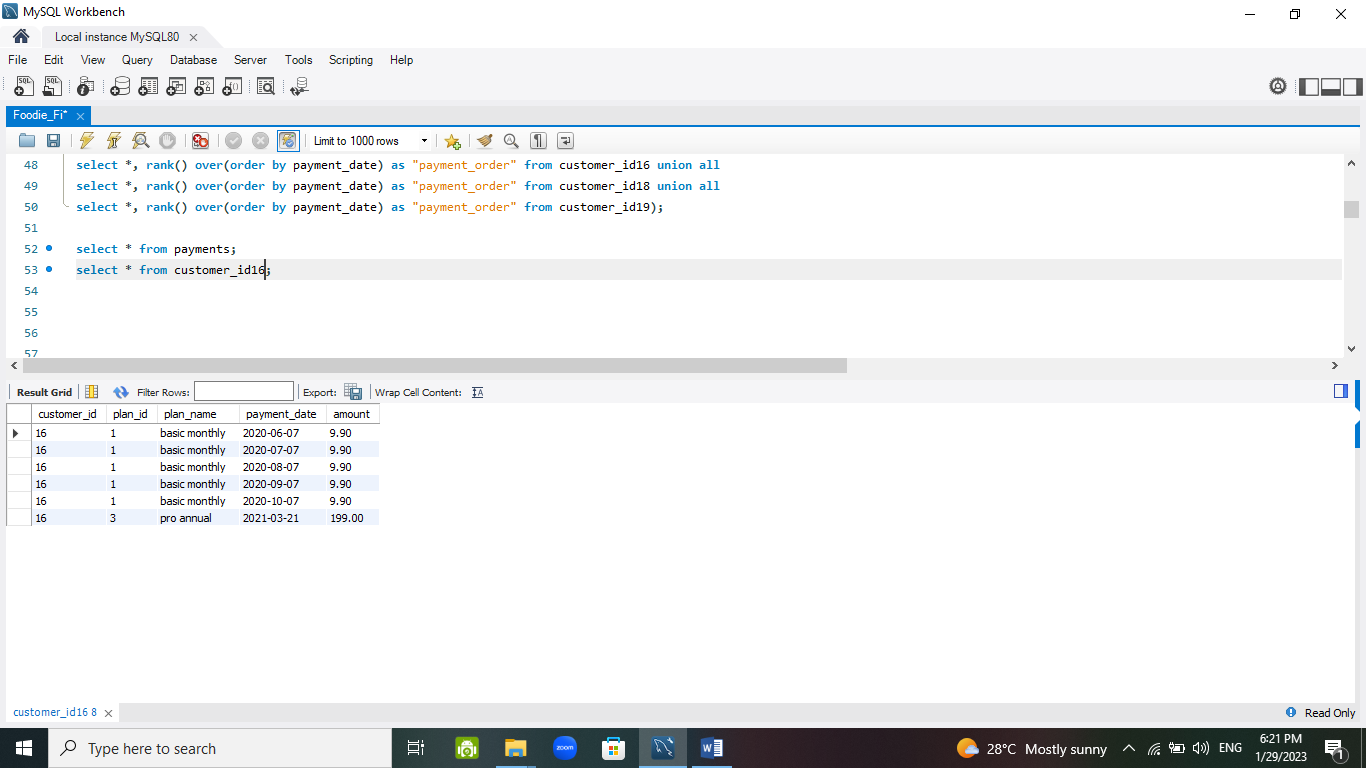
(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 1 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 16 **LIMIT** 1) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 2 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 16 **LIMIT** 1) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 3 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 16 **LIMIT** 1) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 4 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 16 **LIMIT** 1) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 5 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 3 and customer\_id = 16));



------------------------------------------------------------------------------------------------------------------------------------------------

**Customer ID 18:**

**CREATE** view customer\_id18 **AS** (

(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 2 and customer\_id = 18) **UNION ALL**

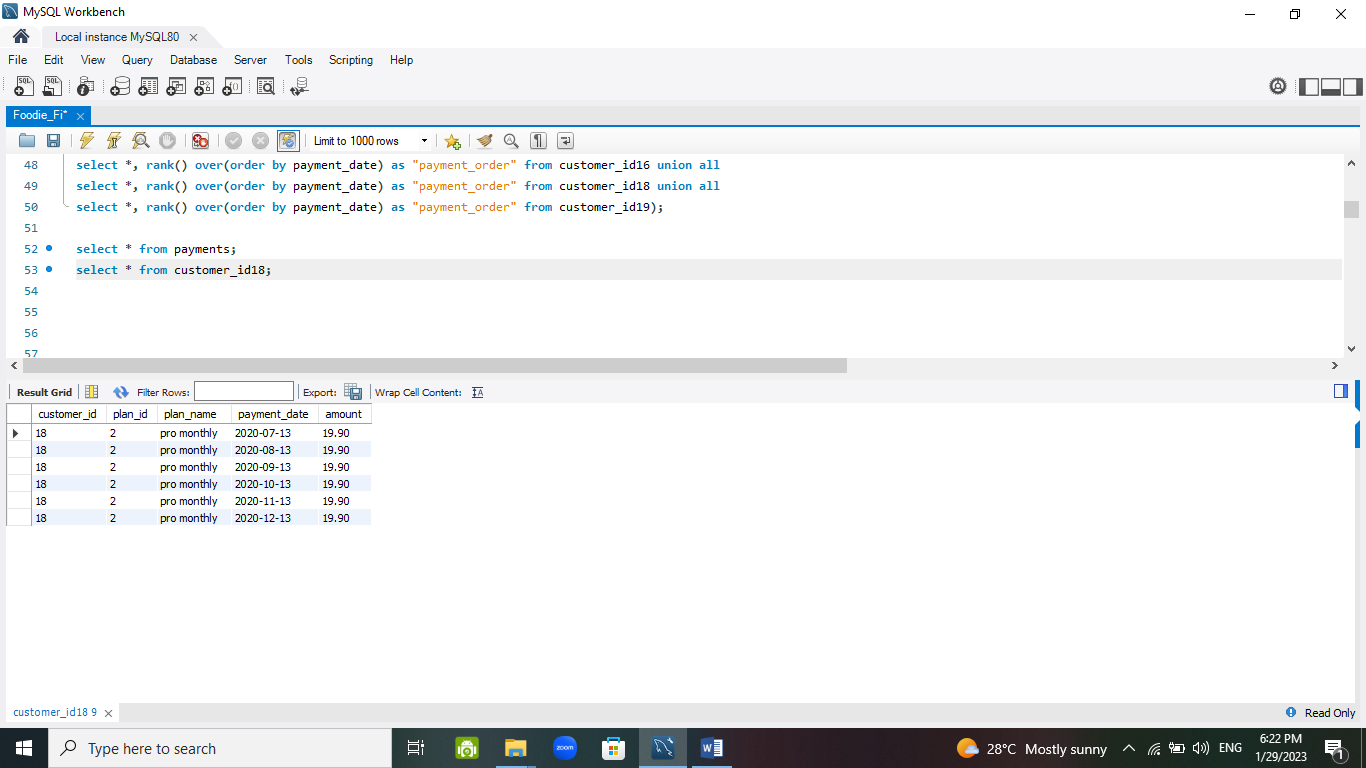
(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 1 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 2 and customer\_id = 18) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 2 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 2 and customer\_id = 18) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 3 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 2 and customer\_id = 18) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 4 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 2 and customer\_id = 18) **UNION ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 5 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id = 2 and customer\_id = 18));



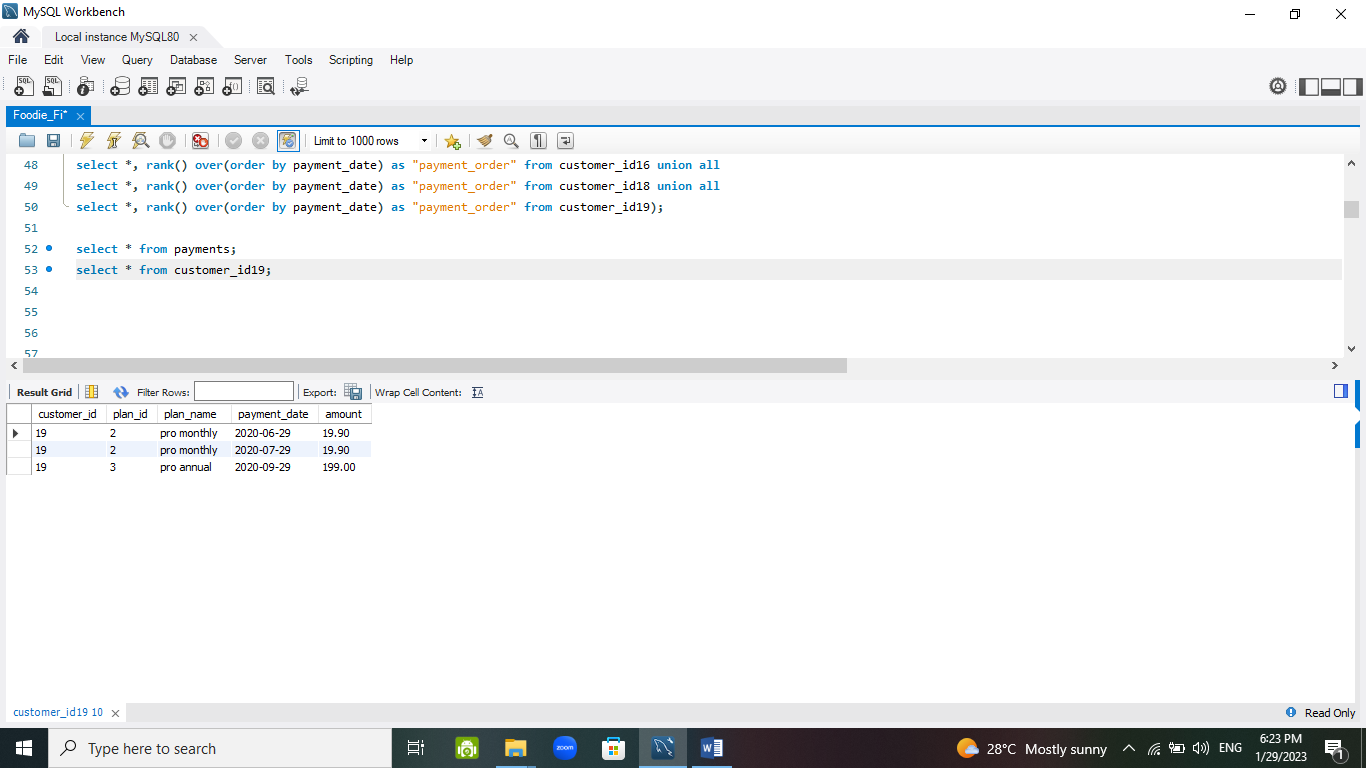
------------------------------------------------------------------------------------------------------------------------------------------------

**Customer ID 19:**

**CREATE** view customer\_id19 **AS** (

(**SELECT** customer\_id, plan\_id, plan\_name, start\_date **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 19 **LIMIT** 1) **UNION** **ALL**

(**SELECT** customer\_id, plan\_id, plan\_name, date\_add(start\_date, interval 1 month) **AS** "payment\_date", price **AS** "amount" **FROM** plans **JOIN** subscriptions using(plan\_id) **WHERE** plan\_id not in (0,4) and customer\_id = 19)) ;



------------------------------------------------------------------------------------------------------------------------------------------------

**CREATE** table payments **AS** (

**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id1 **UNION** **ALL**

**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id2 **UNION** **ALL**

**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id13 **UNION** **ALL**

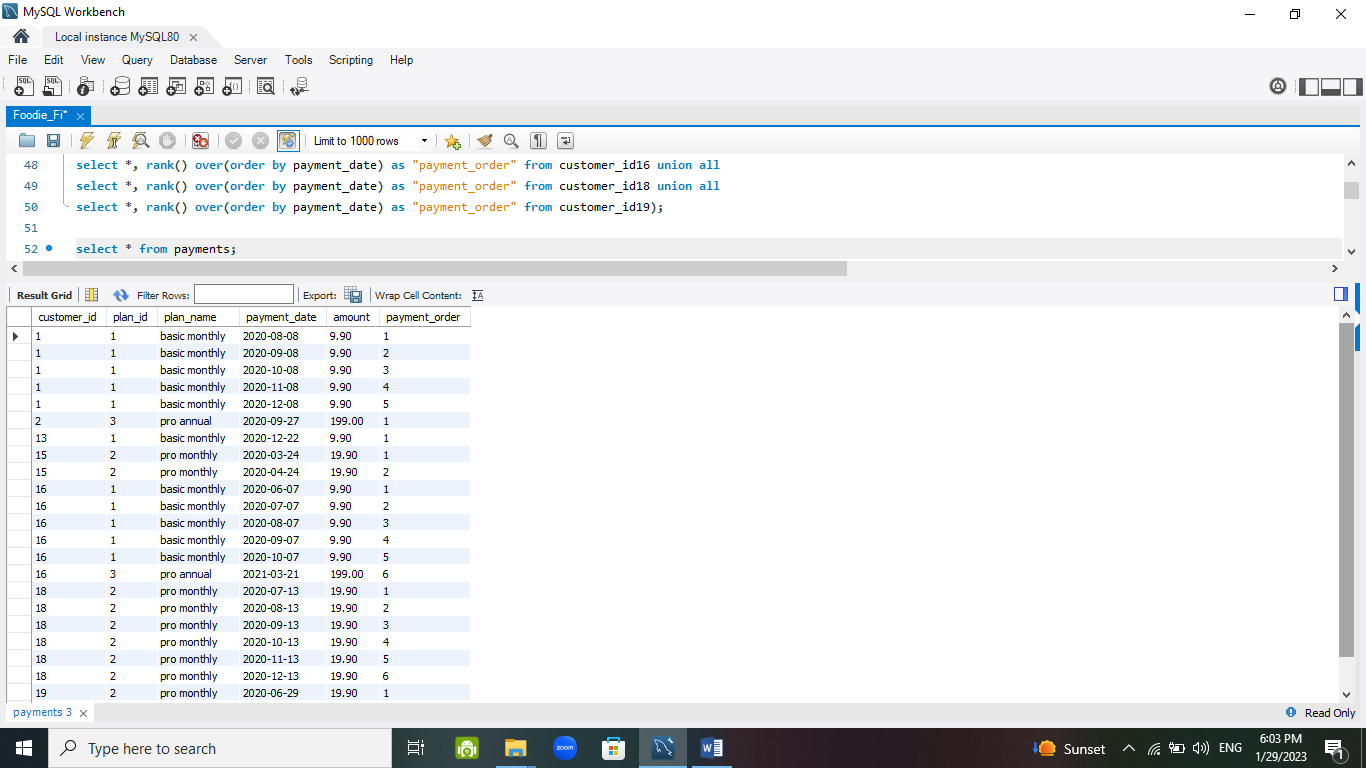
**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id15 **UNION** **ALL**

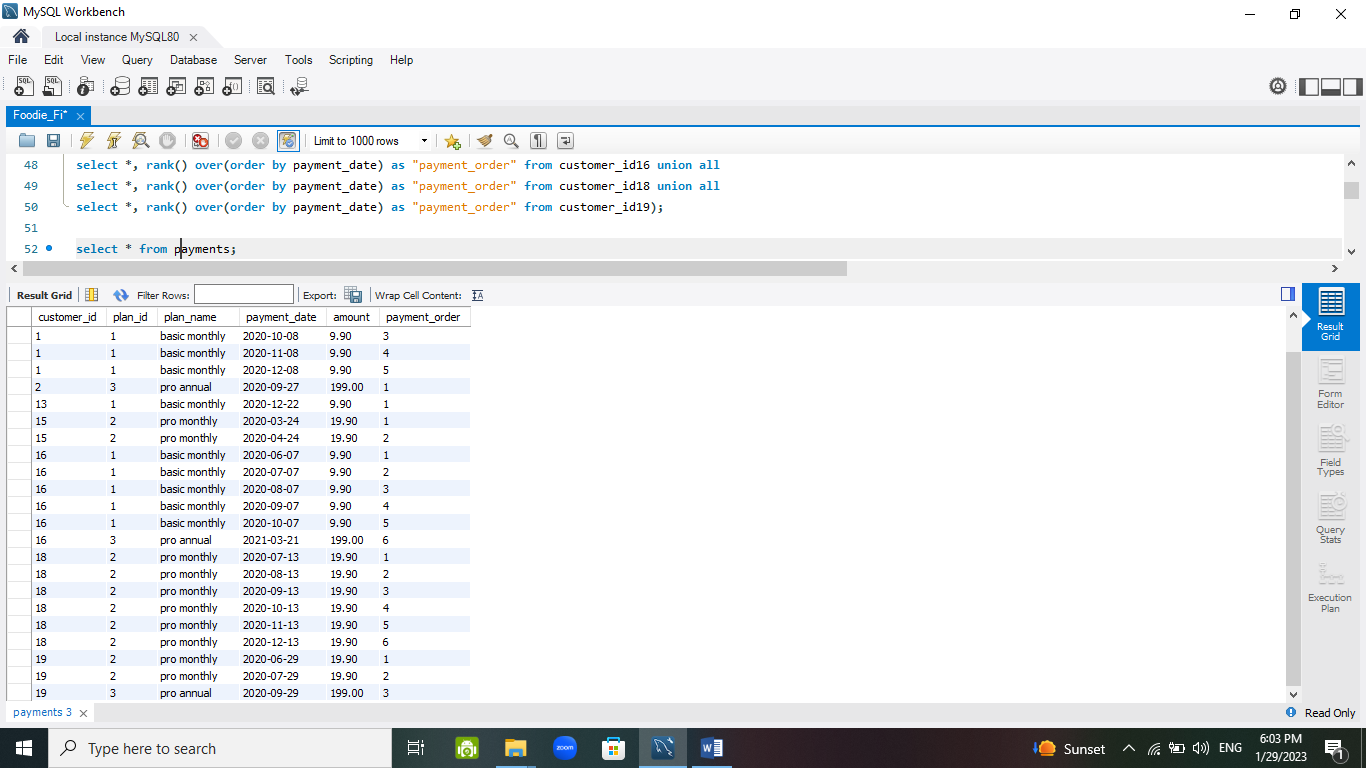
**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id16 **UNION** **ALL**

**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id18 **UNION** **ALL**

**SELECT** \*, rank() over(order by payment\_date) **AS** "payment\_order" **FROM** customer\_id19);

**SELECT** \* **FROM** payments;





1. **Outside the box Questions**

* How would you calculate the rate of growth for Foodie-Fi?

1. No. of customers who have subscribed to the app on the daily or monthly basis.

2. Total Income and Profit can be analysed on the monthly basis and then based on that the gross margin can be calculated.

* What key metrics would you recommend Foodie-Fi management to track over time to assess performance of their overall business?

1. Percentage of Churn / No. of Customers Churned
2. Monthly Revenue generated
3. Number of paying users
4. Retention Rate
5. Profit generated

* What are some key customer journeys or experiences that you would analyse further to improve customer retention?

1. Why the customers are churning the subscription?
2. Best ways to make a subscriber upgrade to an annual plan or higher for making more profit and to generate more revenue.

* If the Foodie-Fi team were to create an exit survey shown to customers who wish to cancel their subscription, what questions would you include in the survey?

1. Rate the services provided by Foodie-Fi (1 to 5)
2. Did you find any content that you liked? (Yes/No)
3. Reason for leaving
4. Feedback

* What business levers could the Foodie-Fi team use to reduce the customer churn rate? How would you validate the effectiveness of your ideas?

Customer Service : Listen to customer’s problems and try to fix it as soon as possible.

Subscribed Customers: Provide some special perks and membership offers to the subscribed customers for the customer’s satisfaction and for also more future subscriptions.

Marketing : Marketing of the app would increase the subscriptions rate significantly, if we collaborate with some recognized personality in the society over different social platforms.