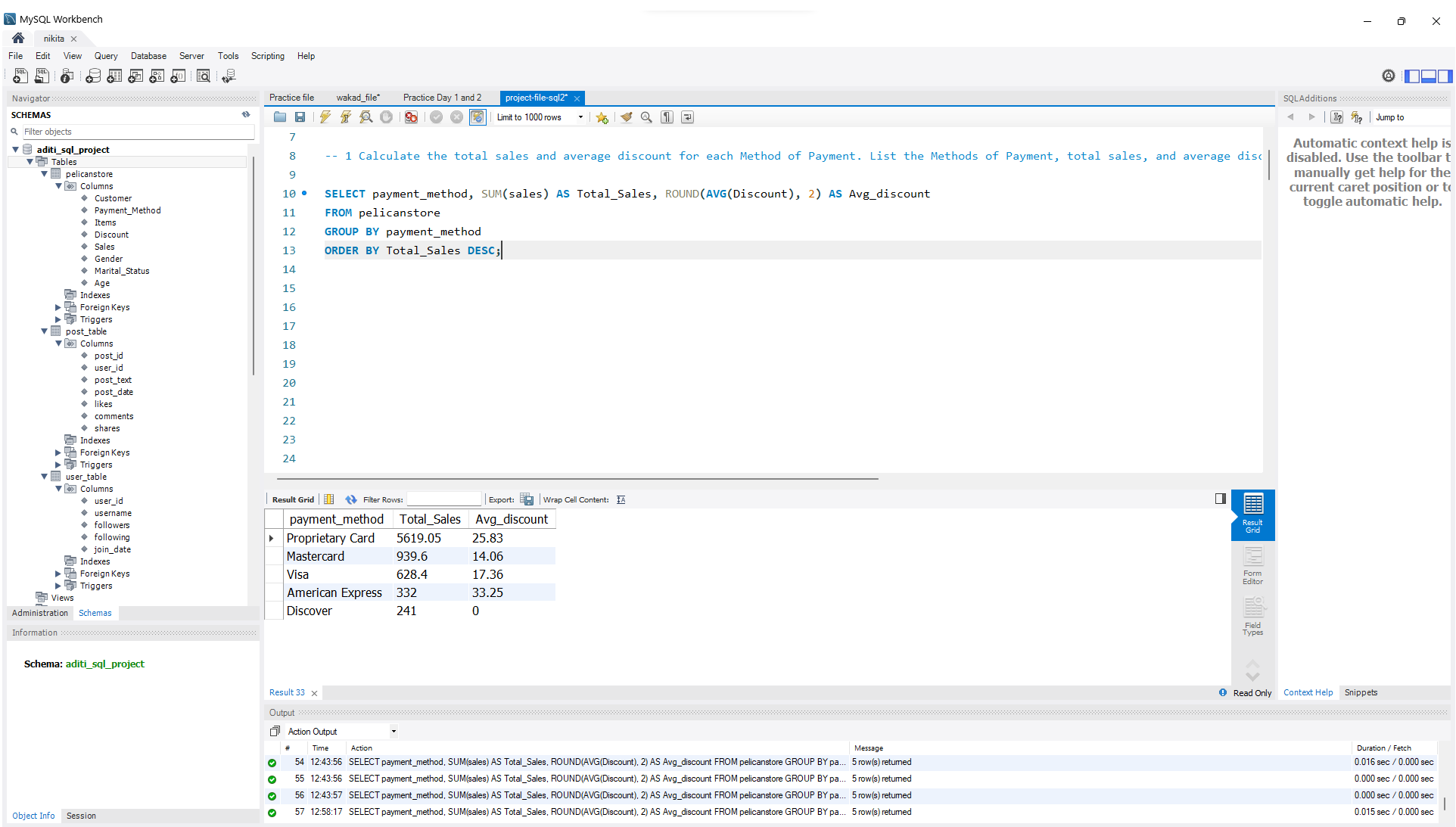
Name – Nikita Gund
Email – nikitagund440@gmail.com
Batch Date – 10/Oct/2023
Course Name – SQL

Database - Pelican Store

Q.1**). Calculate the total sales and average discount for each Method of Payment. List the Methods of Payment, total sales, and average discount in descending order of total sales.**

```
SELECT payment_method, SUM(sales) AS Total_Sales, ROUND(AVG(Discount), 2) AS Avg_discount
FROM pelicanstore
GROUP BY payment_method
ORDER BY Total_Sales DESC;
```



Q2). Identify the age group (e.g., 20-30, 31-40, etc.) with the highest total sales. List the age group and the corresponding total sales.

```
SELECT
   CASE
      WHEN Age BETWEEN 20 AND 40 THEN '20-30'
      WHEN Age BETWEEN 41 AND 60 THEN '31-40'
      WHEN Age BETWEEN 61 AND 80 THEN '41-50'
   END AS Age_Group,
   SUM(Sales) AS Total_Sales
FROM pelicanstore GROUP BY Age_Group ORDER BY Total_Sales DESC LIMIT 1;
```

MySQL Workbench

nikita

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

aditi_sql_project
- Tables
  - pelicanstore
    - Columns: Customer, Payment_Method, Items, Discount, Sales, Gender, Marital_Status, Age
    - Indexes
    - Foreign Keys
    - Triggers
  - post_table
    - Columns: post_id, user_id, post_text, post_date, likes, comments, shares
    - Indexes
    - Foreign Keys
    - Triggers
  - user_table
    - Columns: user_id, username, followers, following, join_date
    - Indexes
    - Foreign Keys
    - Triggers
- Views

Administration Schemas

Information

Schema: aditi_sql_project

Object Info Session

Practice file | wakad_file* | Practice Day 1 and 2 | project-file-sql2*

Limit to 1000 rows

```
13  ORDER BY Total_Sales DESC;
14
15  -- 2 Identify the age group (e.g., 20-30, 31-40, etc.) with the highest total sales. List the age group and the corresponding total sales.
16  SELECT
17      CASE
18          WHEN Age BETWEEN 20 AND 40 THEN '20-30'
19          WHEN Age BETWEEN 41 AND 60 THEN '31-40'
20          WHEN Age BETWEEN 61 AND 80 THEN '41-50'
21      END AS Age_Group,
22      SUM(Sales) AS Total_Sales
23  FROM pelicanstore
24  GROUP BY Age_Group
25  ORDER BY Total_Sales DESC
26  LIMIT 1;
27
28
29
30
```

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

Age_Group	Total_Sales
31-40	3749.77

Result Grid | Form Editor | Field Types

Result 34 | Read Only | Context Help | Snippets

SQLAdditions

Jump to

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
55	12:43:56	SELECT payment_method, SUM(sales) AS Total_Sales, ROUND(AVG(Discount), 2) AS Avg_discount FROM pelicanstore GROUP BY pa...	5 row(s) returned	0.000 sec / 0.000 sec
56	12:43:57	SELECT payment_method, SUM(sales) AS Total_Sales, ROUND(AVG(Discount), 2) AS Avg_discount FROM pelicanstore GROUP BY pa...	5 row(s) returned	0.000 sec / 0.000 sec
57	12:58:17	SELECT payment_method, SUM(sales) AS Total_Sales, ROUND(AVG(Discount), 2) AS Avg_discount FROM pelicanstore GROUP BY pa...	5 row(s) returned	0.015 sec / 0.000 sec
58	12:59:39	SELECT CASE WHEN Age BETWEEN 20 AND 40 THEN '20-30' WHEN Age BETWEEN 41 AND 60 THEN '31-40' W...	1 row(s) returned	0.000 sec / 0.000 sec

Q 3). Find the most common combination of gender and marital status among customers. List the gender, marital status, and the count of customers for the most common combination.

SELECT Gender, Marital_Status, COUNT(Customer) AS Customer_count

FROM pelicanstore

GROUP BY Gender, Marital_Status

ORDER BY Customer_count DESC;

MySQL Workbench

nikita

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

aditi_sql_project
- Tables
  - pelicanstore
    - Columns: Customer, Payment_Method, Items, Discount, Sales, Gender, Marital_Status, Age
    - Indexes
    - Foreign Keys
    - Triggers
  - post_table
    - Columns: post_id, user_id, post_text, post_date, likes, comments, shares
    - Indexes
    - Foreign Keys
    - Triggers
  - user_table
    - Columns: user_id, username, followers, following, join_date
    - Indexes
    - Foreign Keys
    - Triggers
- Views

Administration Schemas

Information

Schema: aditi_sql_project

Object Info Session

Practice file | wakad_file* | Practice Day 1 and 2 | project-file-sql2*

Limit to 1000 rows

```
28
29
30
31  -- 3 Find the most common combination of gender and marital status among customers. List the gender, marital status, and the count of cust
32  SELECT Gender, Marital_Status, COUNT(Customer) AS Customer_count
33  FROM pelicanstore
34  GROUP BY Gender, Marital_Status
35  ORDER BY Customer_count DESC;
36
37
38
39
40
41
42
43
44
45
```

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

Gender	Marital_Status	Customer_count
Female	Married	80
Female	Single	13
Male	Married	4
Male	Single	3

Result Grid | Form Editor | Field Types

Result 35 | Read Only | Context Help | Snippets

SQLAdditions

Jump to

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Output

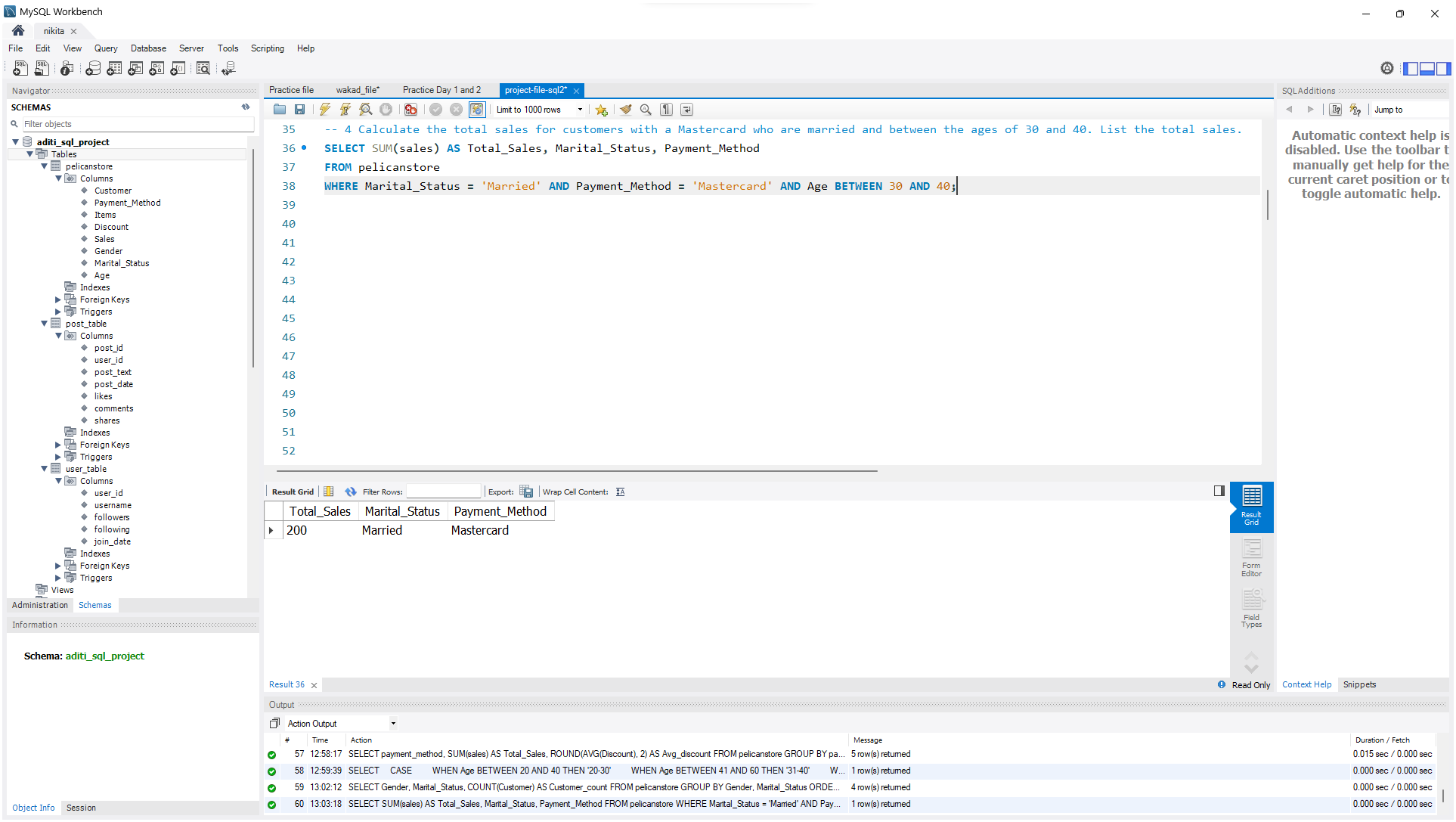
Action Output

#	Time	Action	Message	Duration / Fetch
56	12:43:57	SELECT payment_method, SUM(sales) AS Total_Sales, ROUND(AVG(Discount), 2) AS Avg_discount FROM pelicanstore GROUP BY pa...	5 row(s) returned	0.000 sec / 0.000 sec
57	12:58:17	SELECT payment_method, SUM(sales) AS Total_Sales, ROUND(AVG(Discount), 2) AS Avg_discount FROM pelicanstore GROUP BY pa...	5 row(s) returned	0.015 sec / 0.000 sec
58	12:59:39	SELECT CASE WHEN Age BETWEEN 20 AND 40 THEN '20-30' WHEN Age BETWEEN 41 AND 60 THEN '31-40' W...	1 row(s) returned	0.000 sec / 0.000 sec
59	13:02:12	SELECT Gender, Marital_Status, COUNT(Customer) AS Customer_count FROM pelicanstore GROUP BY Gender, Marital_Status ORDE...	4 row(s) returned	0.000 sec / 0.000 sec

Q 4). Calculate the total sales for customers with a Mastercard who are married and between the ages of 30 and 40. List the total sales.

SELECT SUM(sales) AS Total_Sales, Marital_Status, Payment_Method FROM pelicanstore

WHERE Marital_Status = 'Married' AND Payment_Method = 'Mastercard' AND Age BETWEEN 30 AND 40;

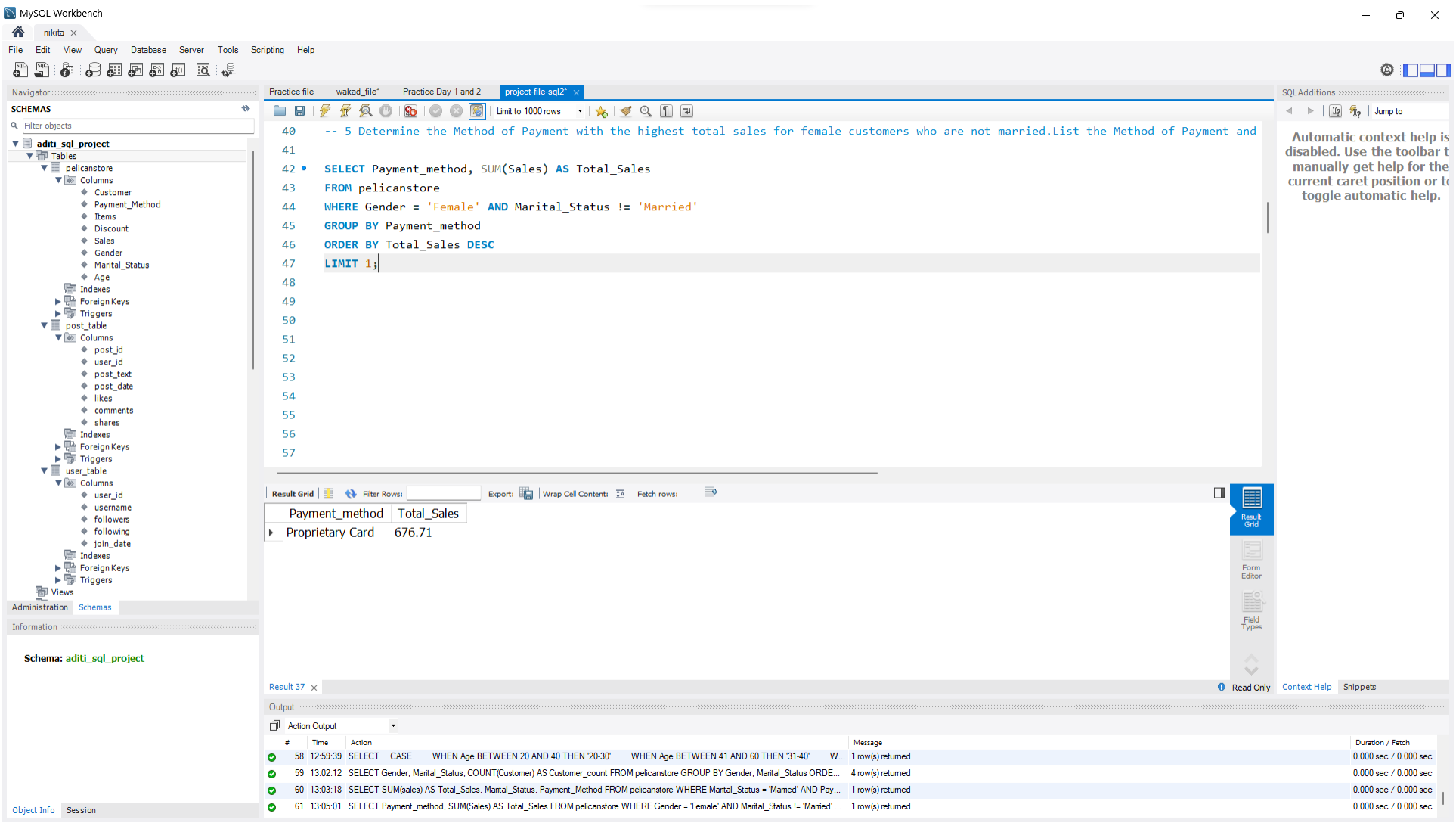


Q 5). Determine the Method of Payment with the highest total sales for female customers who are not married.List the Method of Payment and the total sales.

SELECT Payment_method, SUM(Sales) AS Total_Sales

FROM pelicanstore WHERE Gender = 'Female' AND Marital_Status != 'Married'

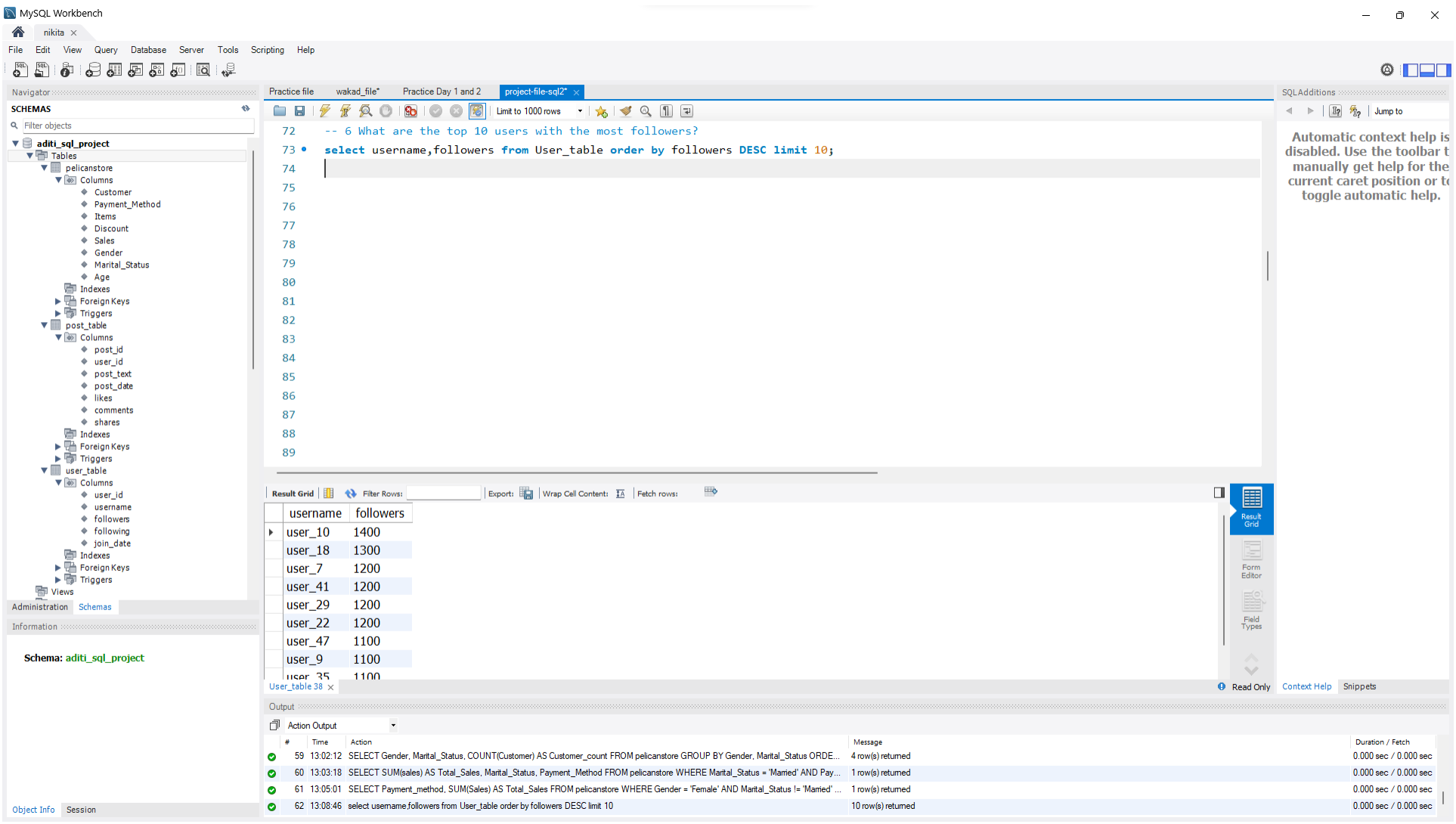
GROUP BY Payment_method ORDER BY Total_Sales DESC LIMIT 1;



Database – User_Table and Post_Table(social media analysis)

Q6). What are the top 10 users with the most followers?

select username,followers from User_table order by followers DESC limit 10;



Q7). Which user has the highest number of likes?

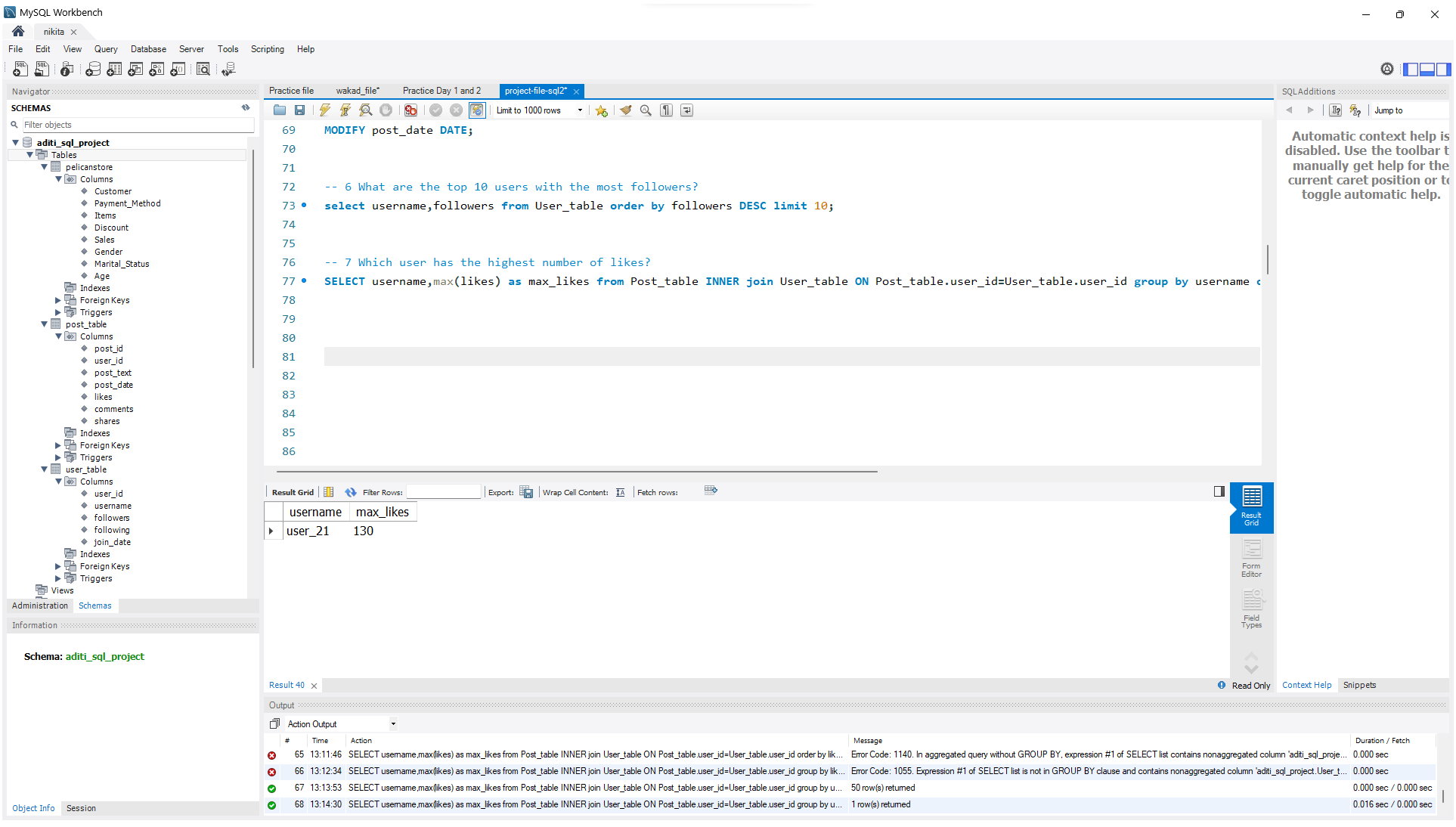
SELECT username,max(likes) as max_likes

from Post_table INNER join User_table ON Post_table.user_id=User_table.user_id

group by username

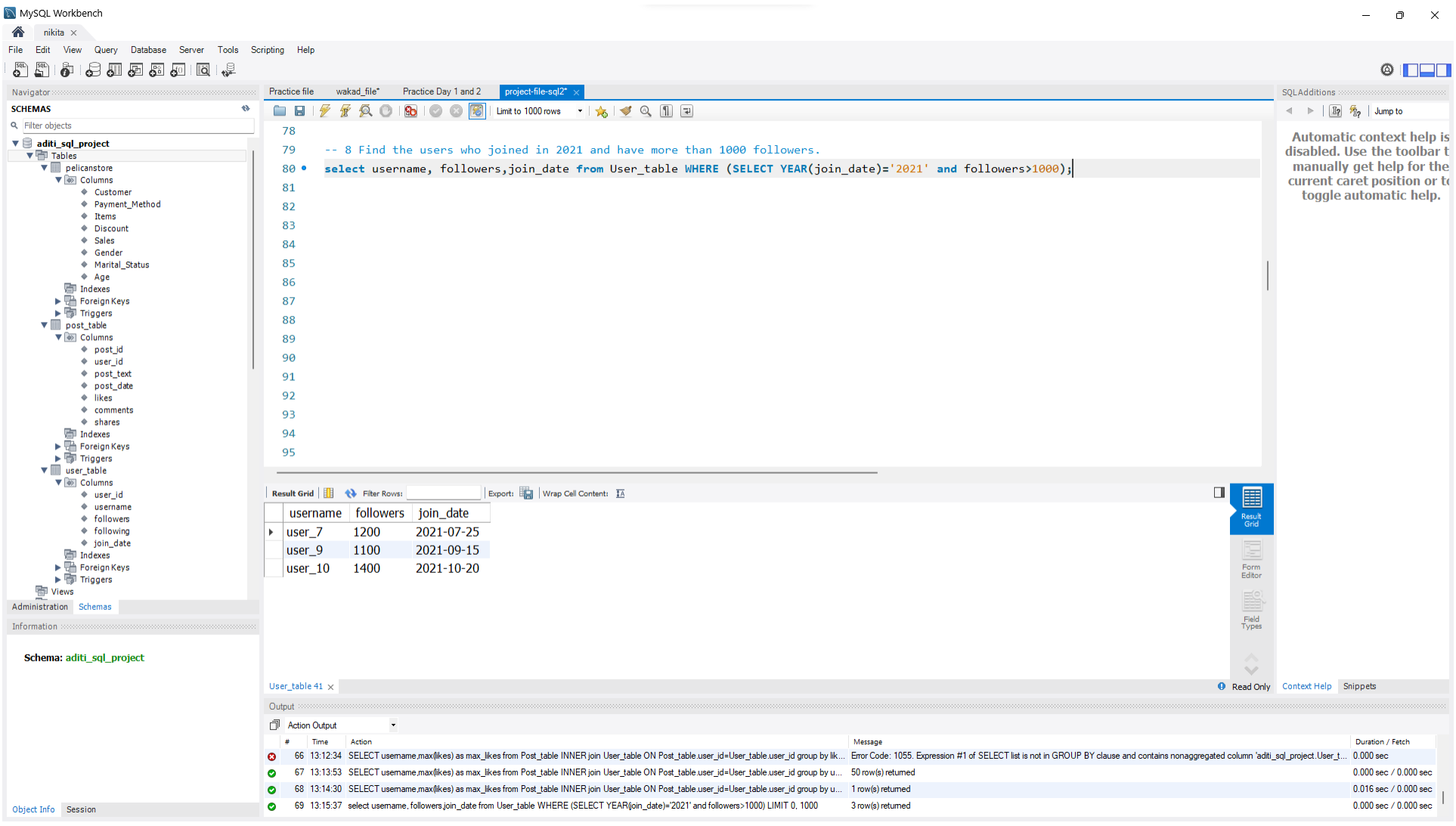
order by max_likes desc

limit 1;



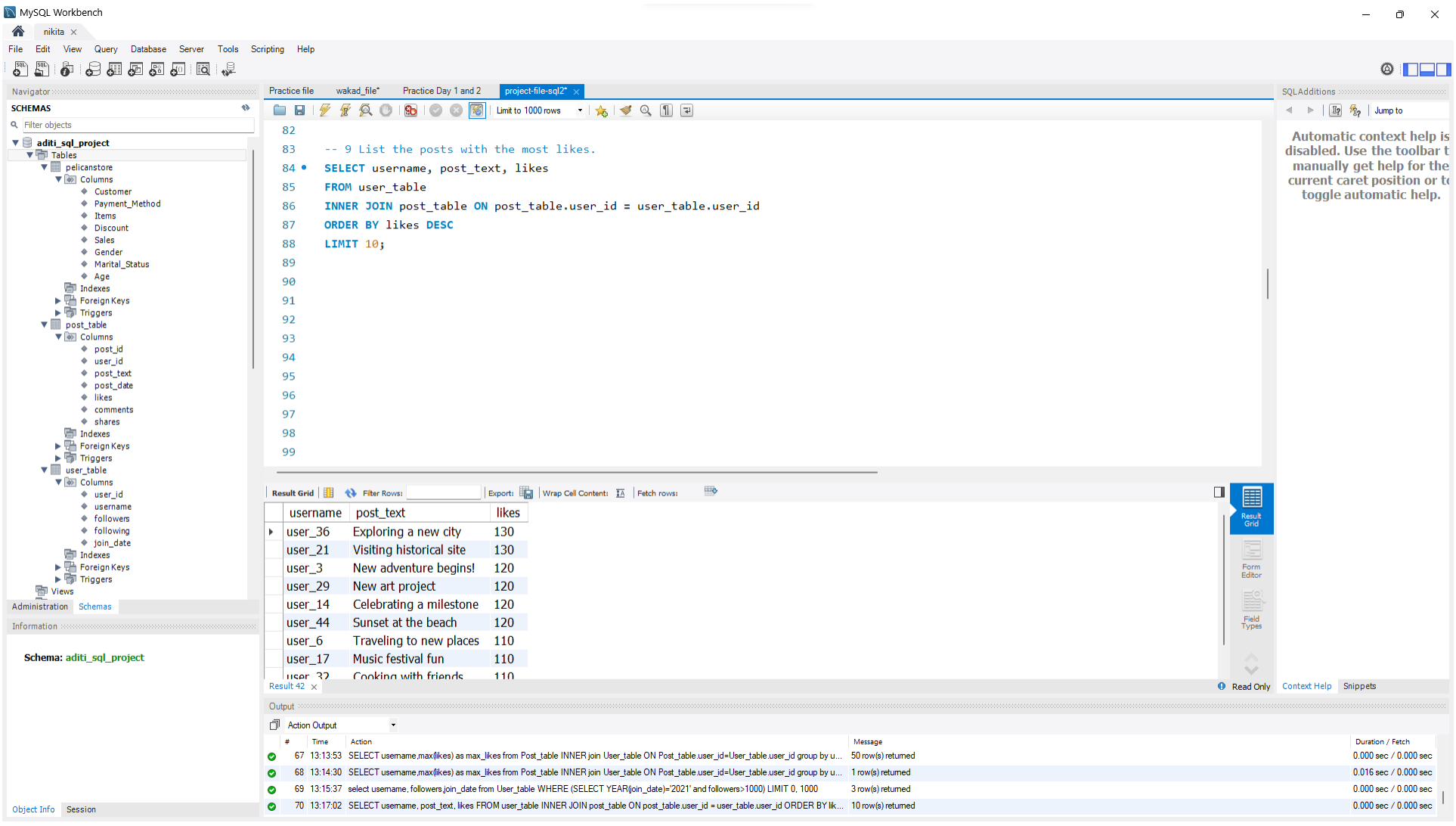
Q8). Find the users who joined in 2021 and have more than 1000 followers.

```
select username, followers,join_date

from User_table WHERE (SELECT YEAR(join_date)='2021' and followers>1000);
```



Q 9). List the posts with the most likes.

```
SELECT username, post_text, likes

 FROM user_table

INNER JOIN post_table ON post_table.user_id = user_table.user_id

ORDER BY likes DESC

 LIMIT 10;
```

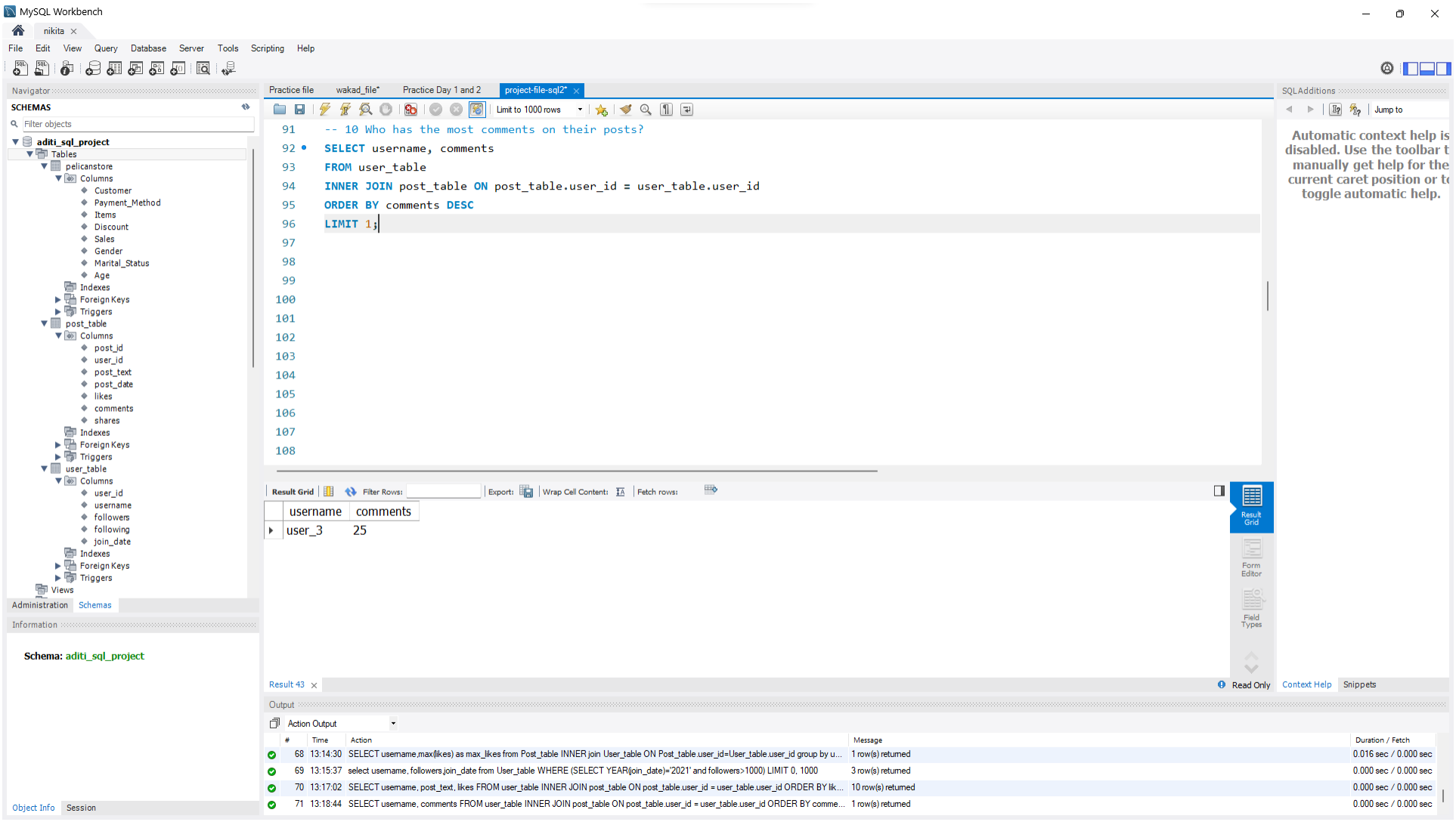


Q 10). Who has the most comments on their posts?

SELECT username, comments FROM user_table

INNER JOIN post_table ON post_table.user_id = user_table.user_id

ORDER BY comments DESC LIMIT 1;

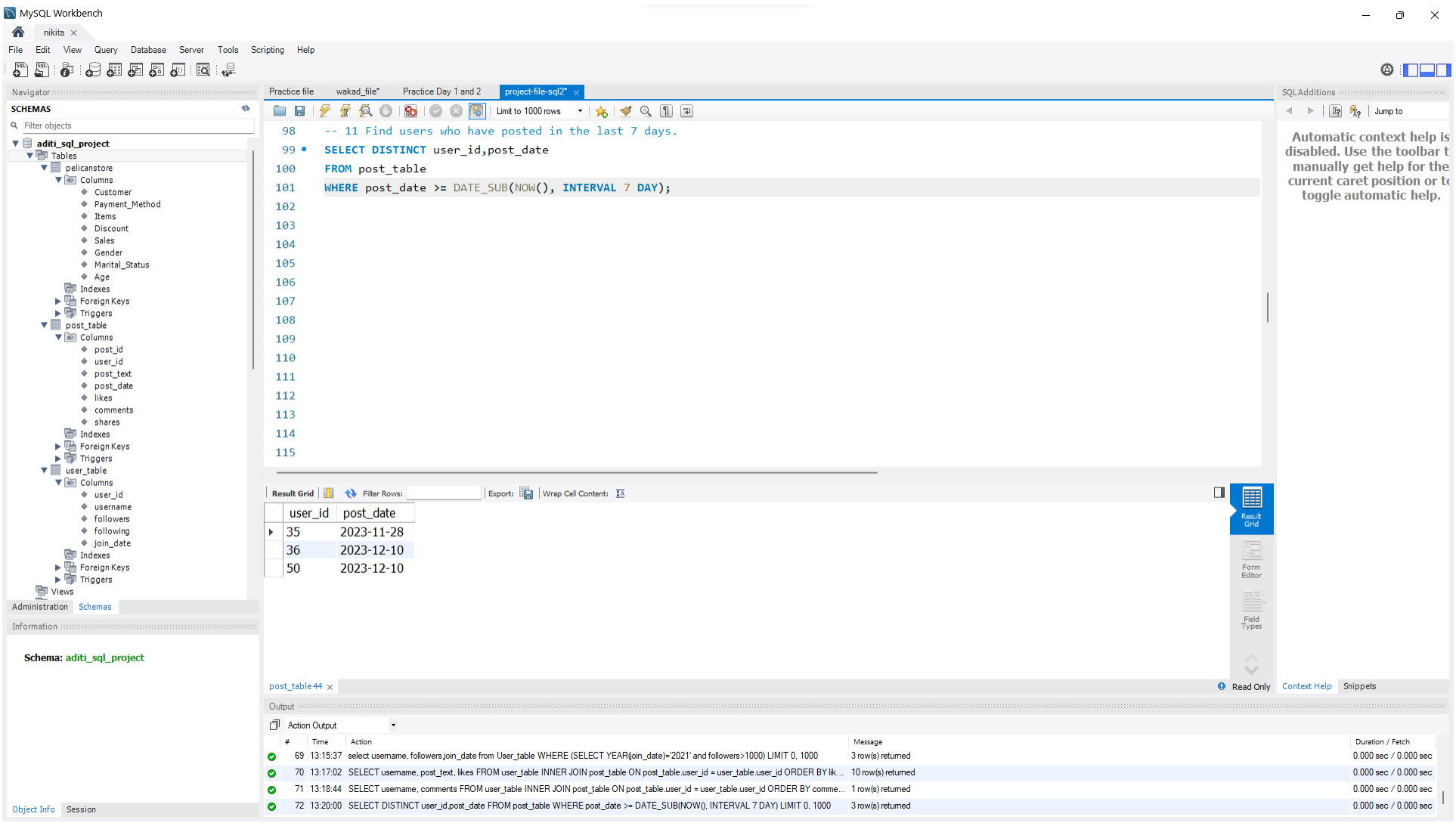


Q11). Find users who have posted in the last 7 days.

SELECT DISTINCT user_id,post_date

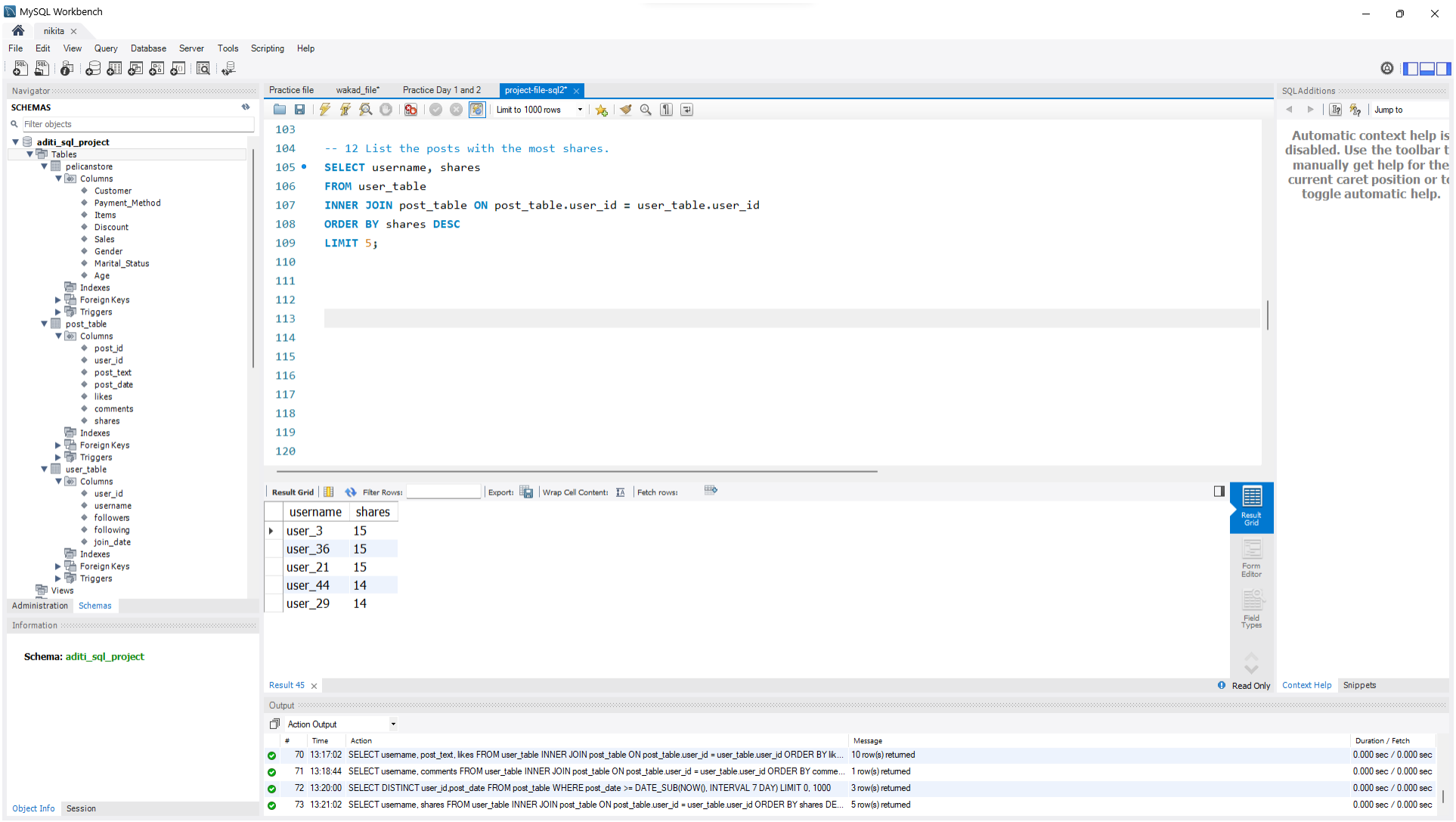
FROM post_table

WHERE post_date >= DATE_SUB(NOW(), INTERVAL 7 DAY);



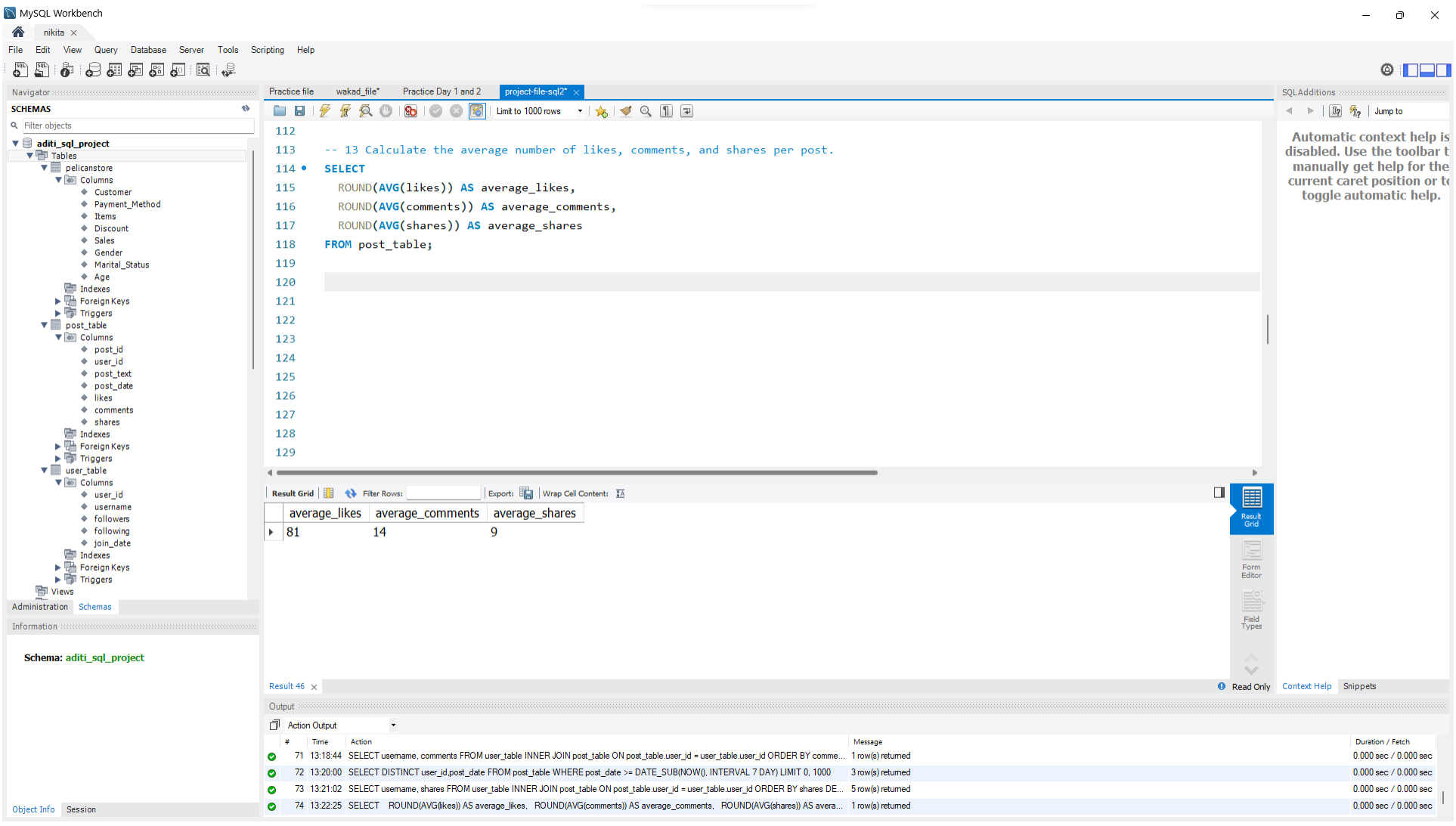
Q 12). List the posts with the most shares.

```
SELECT username, shares FROM user_table
INNER JOIN post_table ON post_table.user_id = user_table.user_id
ORDER BY shares DESC LIMIT 5;
```



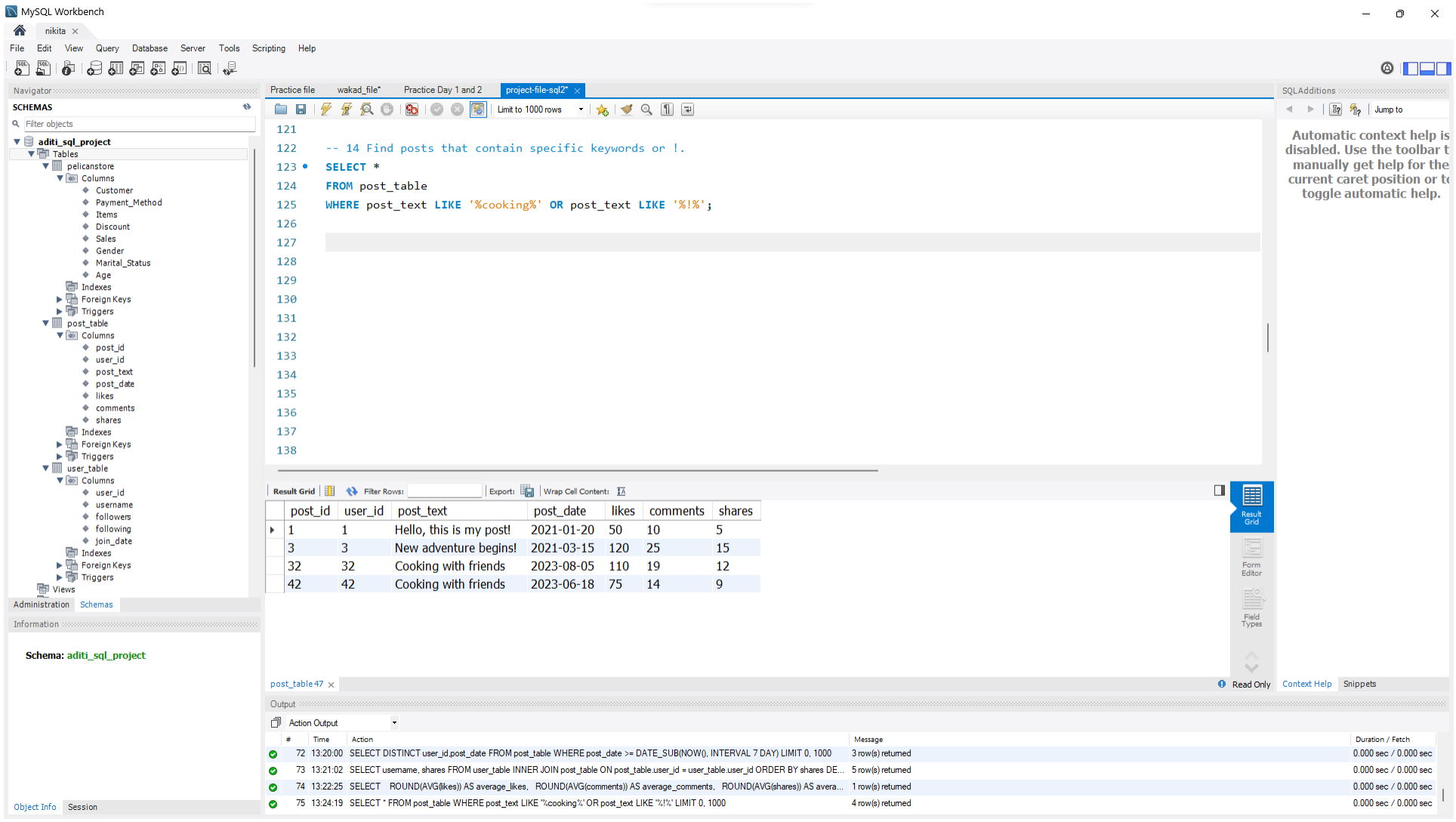
Q13). Calculate the average number of likes, comments, and shares per post.

```
SELECT
  ROUND(AVG(likes)) AS average_likes,
  ROUND(AVG(comments)) AS average_comments,
  ROUND(AVG(shares)) AS average_shares
FROM post_table;
```



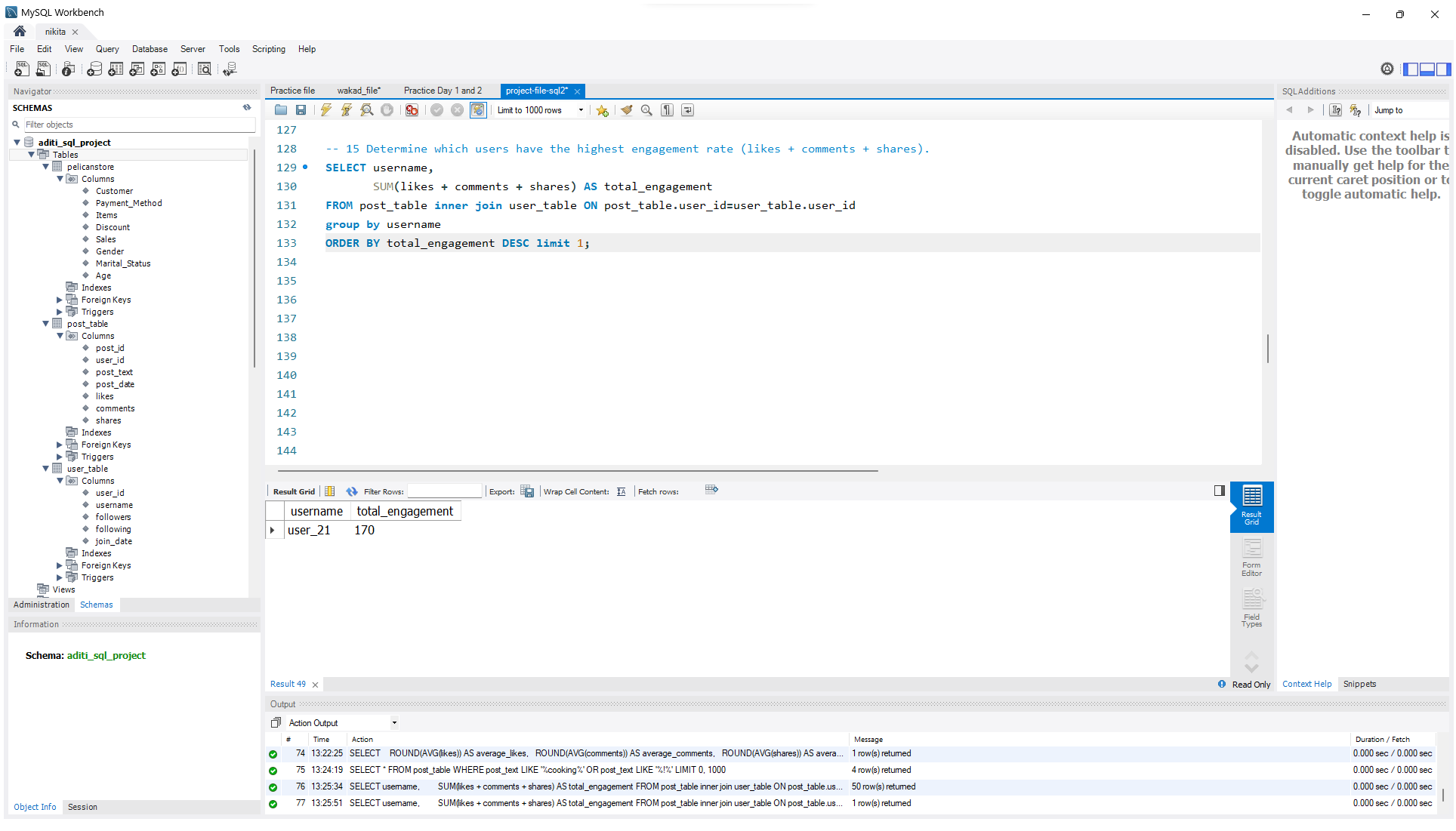
Q14). Find posts that contain specific keywords or !.

```
SELECT *

FROM post_table

WHERE post_text LIKE '%cooking%' OR post_text LIKE '%!%';
```



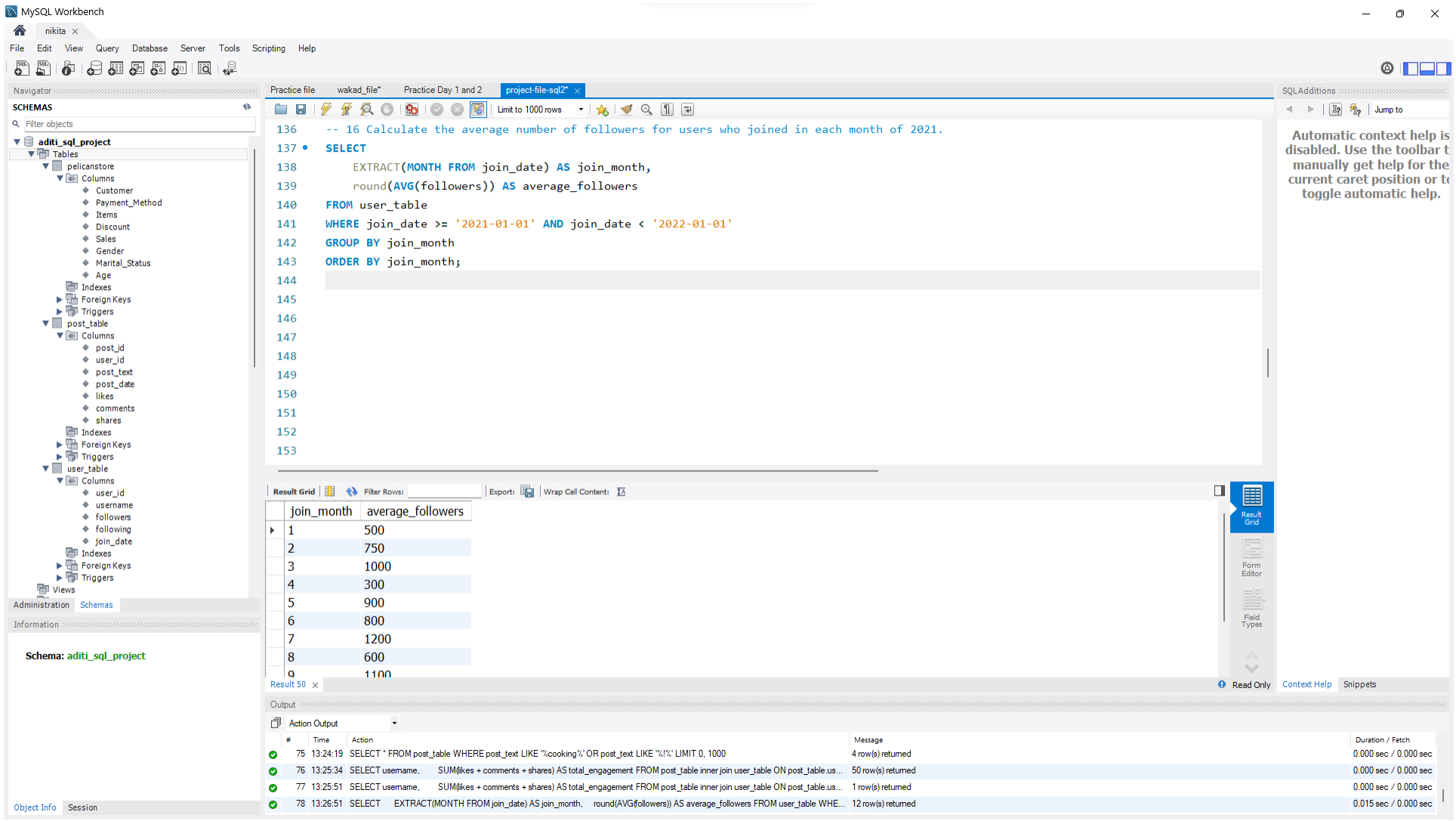
Q 15). Determine which users have the highest engagement rate (likes + comments + shares).

```
SELECT username,
      SUM(likes + comments + shares) AS total_engagement
FROM post_table inner join user_table ON post_table.user_id=user_table.user_id
group by username
ORDER BY total_engagement DESC limit 1;
```



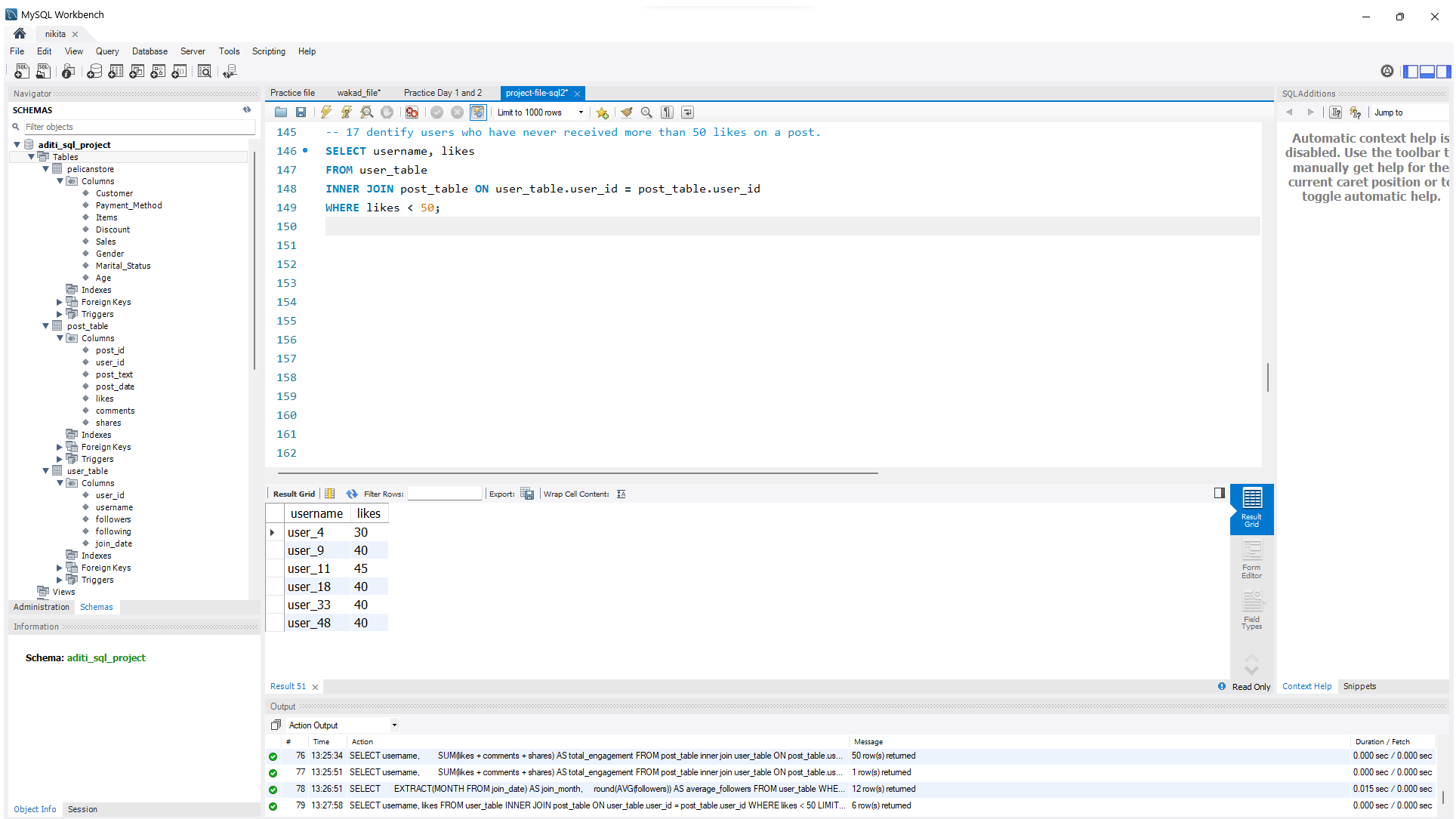
Q16). Calculate the average number of followers for users who joined in each month of 2021.

```
SELECT
    EXTRACT(MONTH FROM join_date) AS join_month,
    round(AVG(followers)) AS average_followers FROM user_table
WHERE join_date >= '2021-01-01' AND join_date < '2022-01-01' GROUP BY join_month ORDER BY join_month;
```



Q17). dentify users who have never received more than 50 likes on a post.

```
SELECT username, likes FROM user_table

INNER JOIN post_table ON user_table.user_id = post_table.user_id WHERE likes < 50;
```

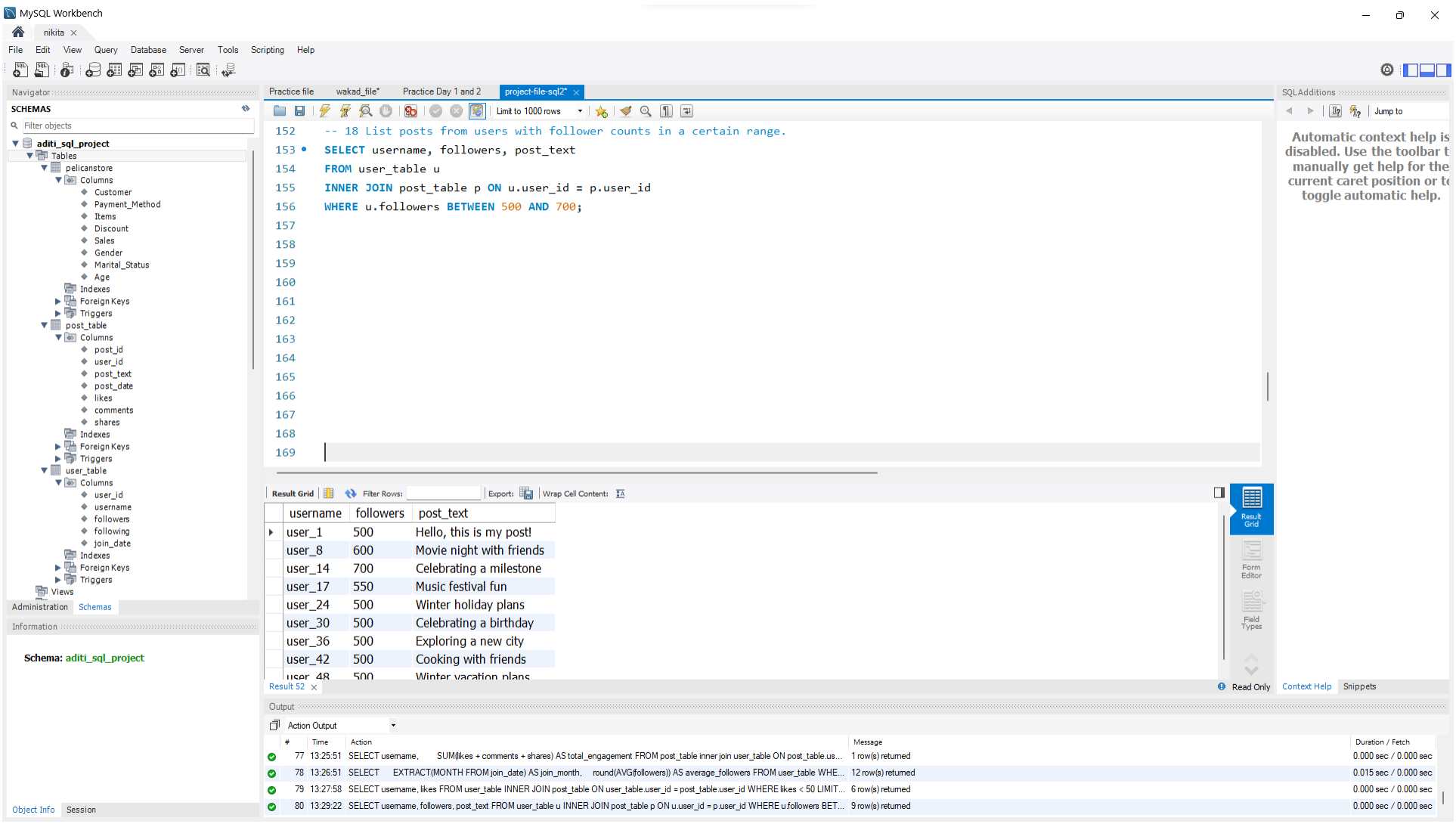


Q18). List posts from users with follower counts in a certain range.

SELECT username, followers, post_text FROM user_table u

INNER JOIN post_table p ON u.user_id = p.user_id

WHERE u.followers BETWEEN 500 AND 700;

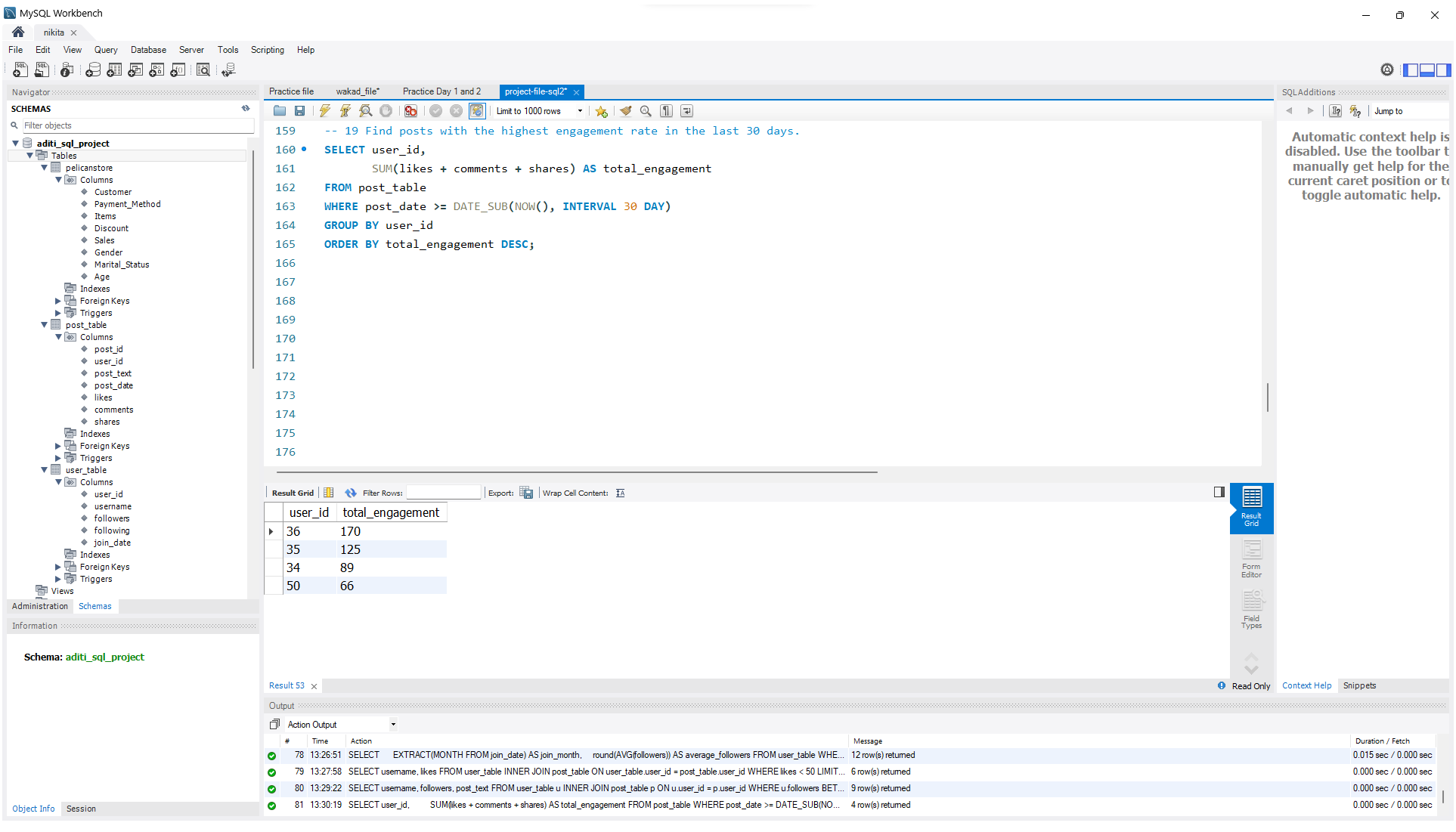


Q19). Find posts with the highest engagement rate in the last 30 days.

SELECT user_id, SUM(likes + comments + shares) AS total_engagement FROM post_table

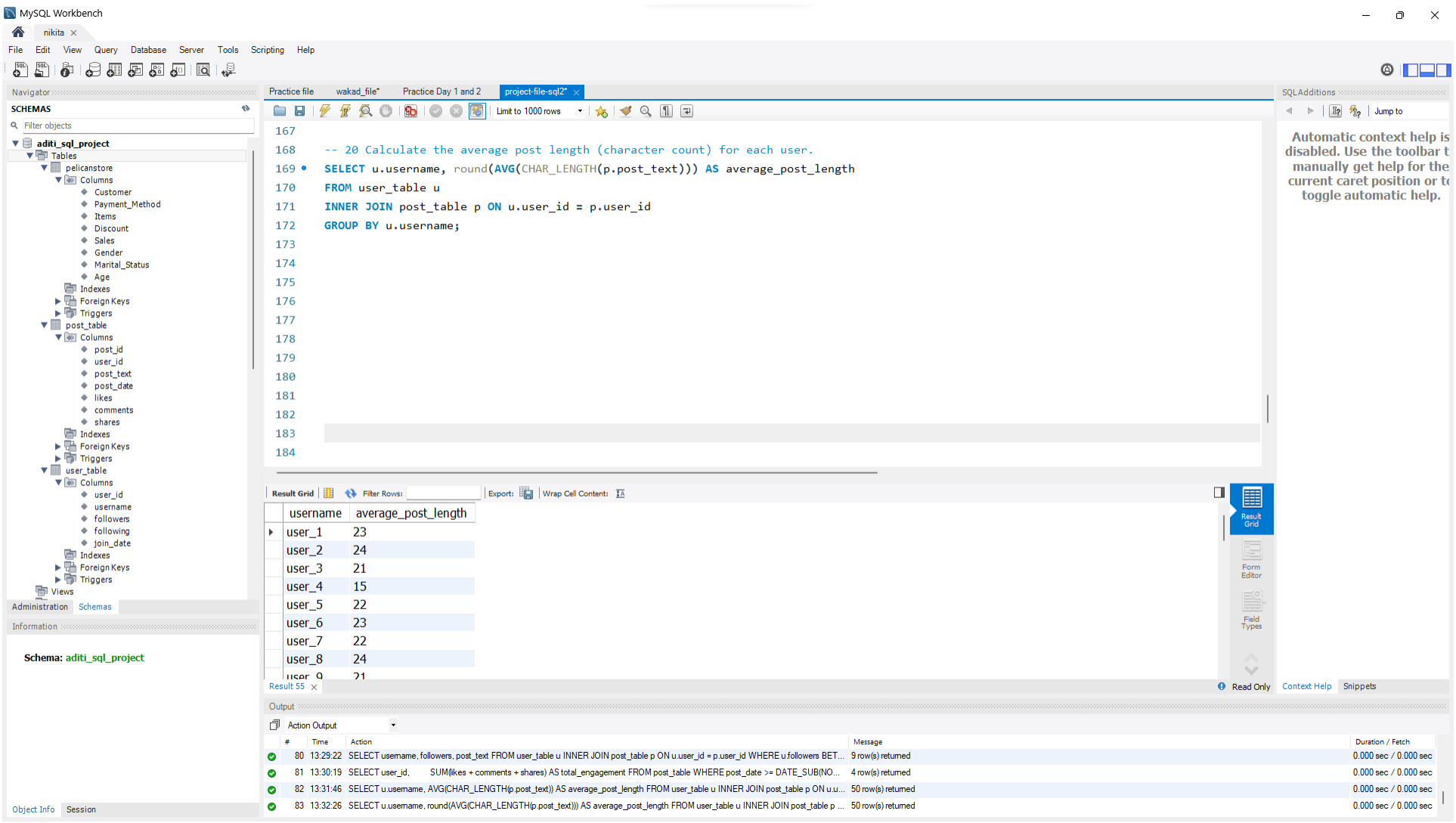
WHERE post_date >= DATE_SUB(NOW(), INTERVAL 30 DAY) GROUP BY user_id

ORDER BY total_engagement DESC;



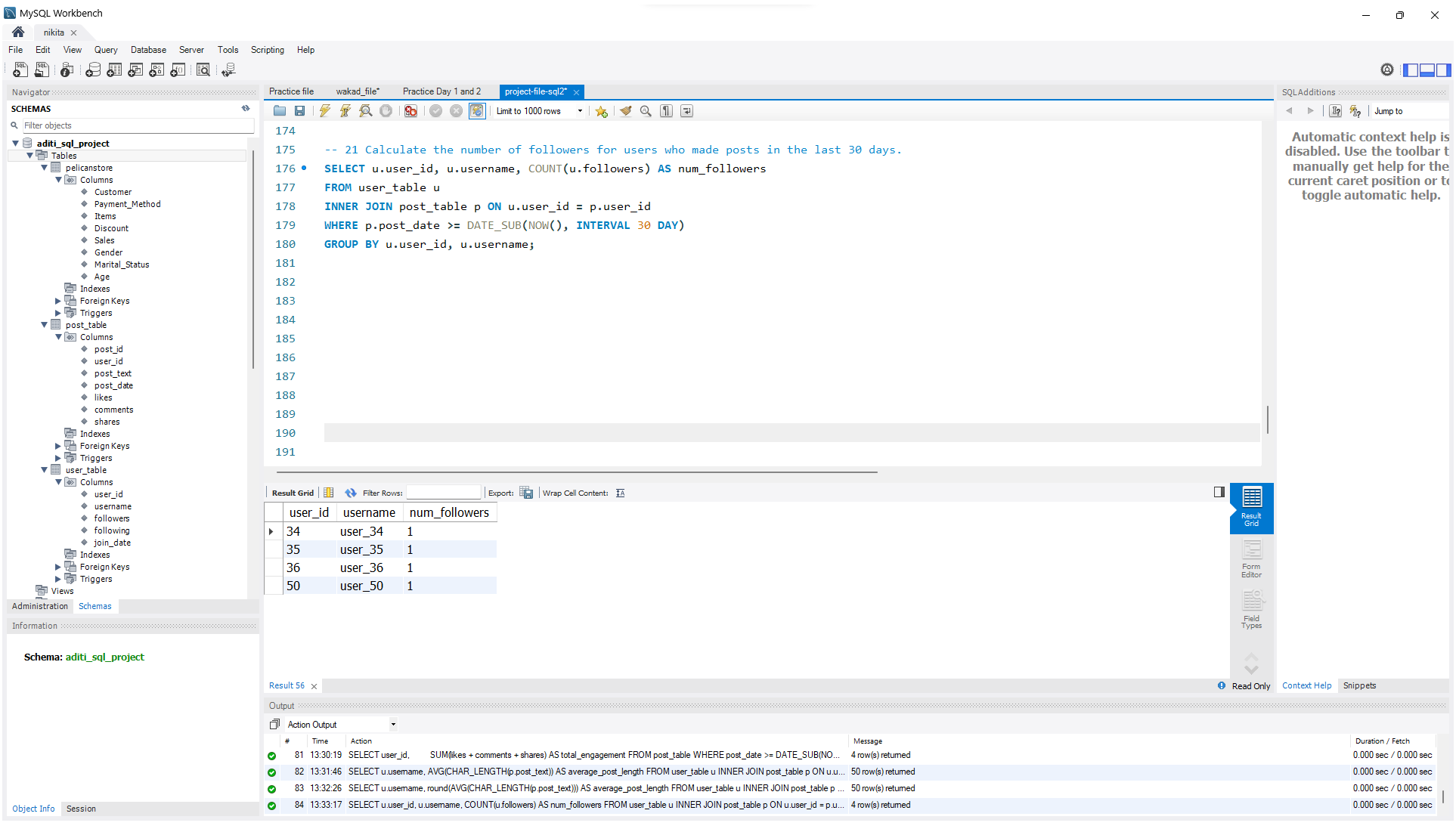
Q 20). Calculate the average post length (character count) for each user.

```
SELECT u.username, round(AVG(CHAR_LENGTH(p.post_text))) AS average_post_length
FROM user_table u
INNER JOIN post_table p ON u.user_id = p.user_id
GROUP BY u.username;
```



Q 21). Calculate the number of followers for users who made posts in the last 30 days.

```
SELECT u.user_id, u.username, COUNT(u.followers) AS num_followers FROM user_table u
INNER JOIN post_table p ON u.user_id = p.user_id
WHERE p.post_date >= DATE_SUB(NOW(), INTERVAL 30 DAY)
GROUP BY u.user_id, u.username;
```



Q22). List posts made by users who have joined the platform in the same month and year. Include the usernames of both the users who made the posts and the join dates of those users.SELECT U.username, P.post_text

SELECT U.username AS user_username, P.post_text, U.join_date

FROM User_table U

JOIN Post_table P ON U.user_id = P.user_id

WHERE DATE_FORMAT(U.join_date, '%Y-%m') = DATE_FORMAT(P.post_date, '%Y-%m');