Instagram User Analytics

by Suraj Beloshe

Powered by

trainity

Project Description

1. Data collected from various Instagram users are given for analysis
2. Need to prepare Database using SQL
3. With help of this data need to find the following insights which will help the Marketing team and Investor metrics.

Marketing

- Find the most loyal users
- Find inactive users
- Find contest winner
- Appealing HASHTAG
- Weekday to launch campaign

Investor Metrics

- User Engagement
- Bots & Fake Accounts

Approach

The data given is large and cannot be handled and couldn't analyze in MS Excel easily. So by using My SQL workbench will write quarries to get desired output. First of all, will create Database from the given data. Will check what kind of tables are available and then will try to build quarries to active answers to questions asked by Business.

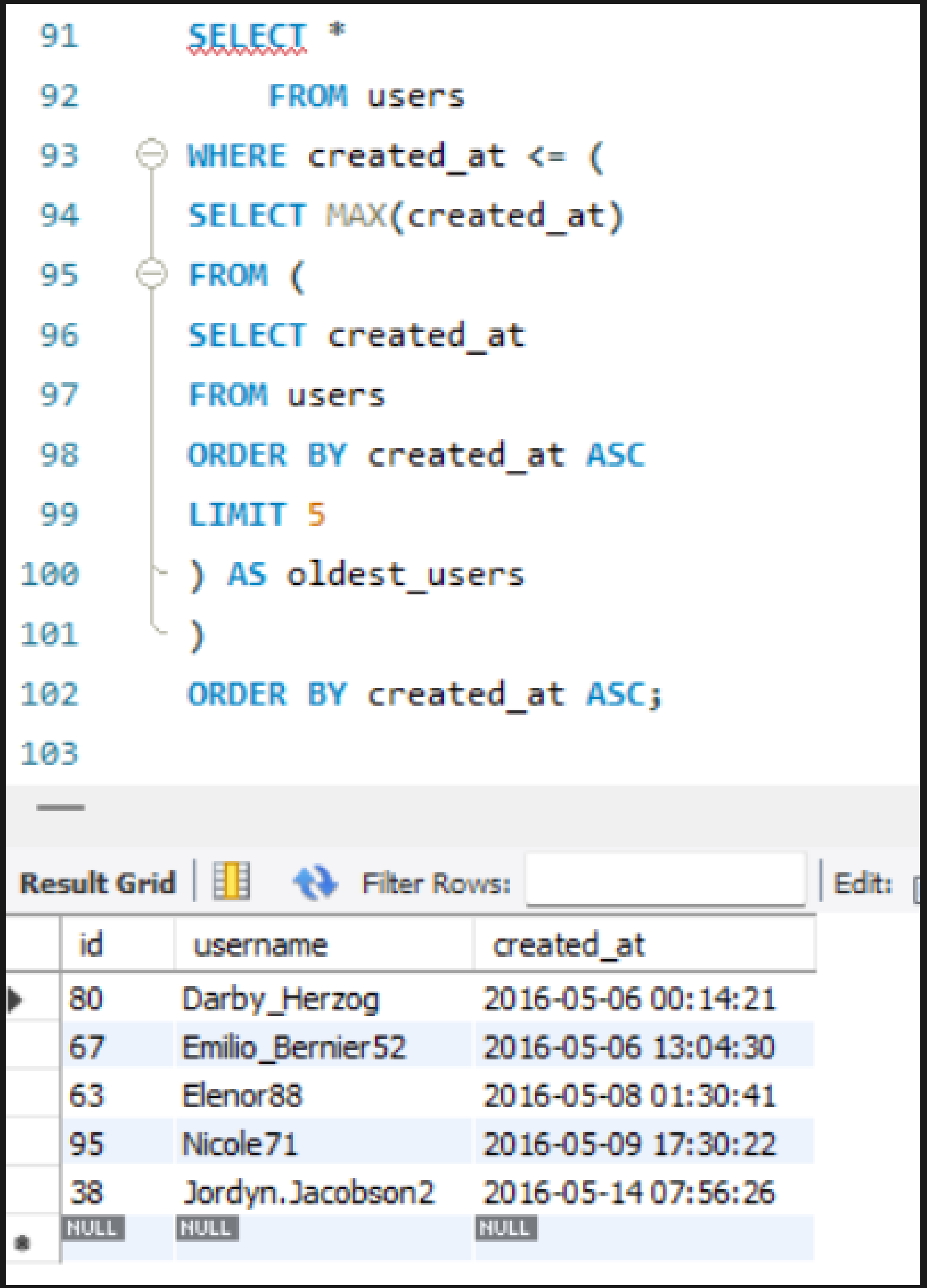
Tech-Stack Used

My SQL workbench version 8.0.32

- MySQL is one of the most popular and widely used SQL databases.
- 2. User-friendly tool for data analysts to work with databases
- 3. It includes a visual SQL query builder, built-in data visualization tools, and collaboration features with other stakeholders.
- 4. The tool also integrates with other commonly used data analysis tools such as Python and R

Insights

Find the 5 oldest users of the Instagram from the database provided



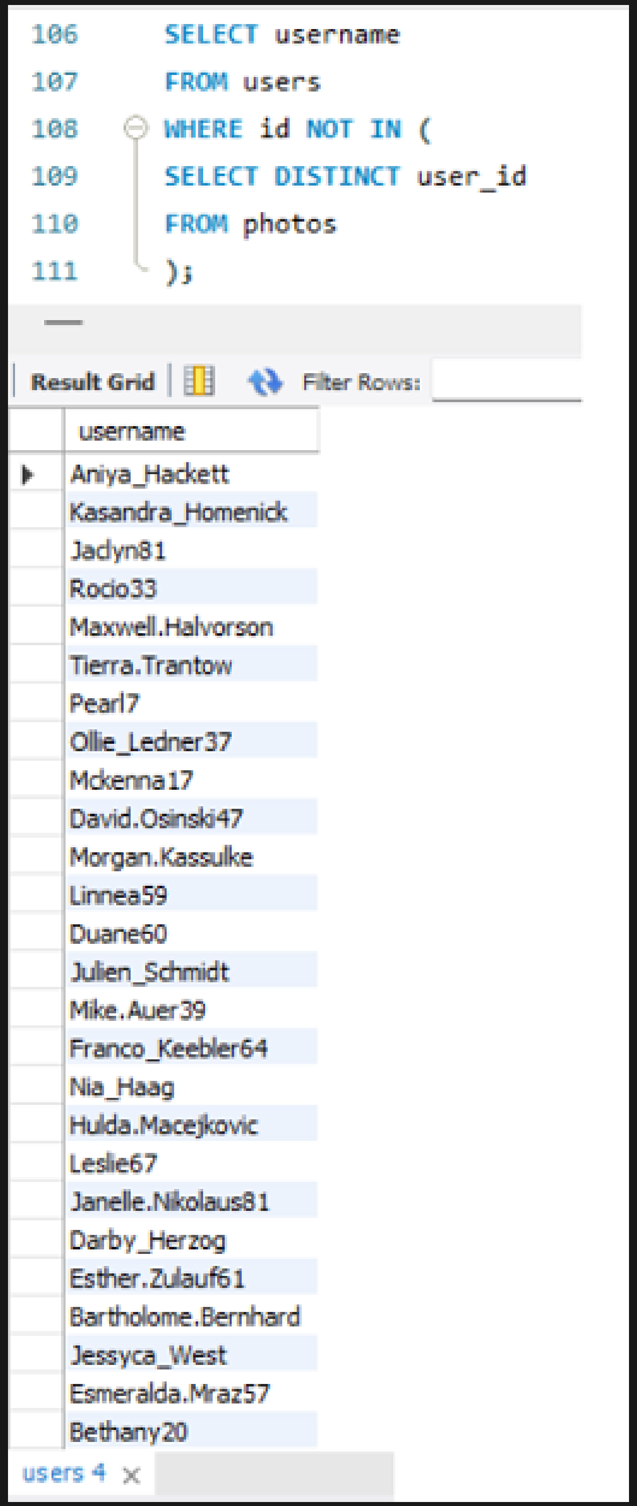
```
SELECT *
    FROM users
WHERE created_at <= (
SELECT MAX(created_at)
FROM (
SELECT created_at
FROM users
ORDER BY created_at ASC
LIMIT 5
) AS oldest_users
)
ORDER BY created_at ASC;
```

id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

- we had a table, with the user id and account creation date
- First, select the created_at column from the user's table and order by using asc further limited to 5 accounts so we get the five oldest accounts.
- we alias it as oldest_users
- we filtered out this result by using a subquery

Insights

Find the users who have never posted a single photo on Instagram



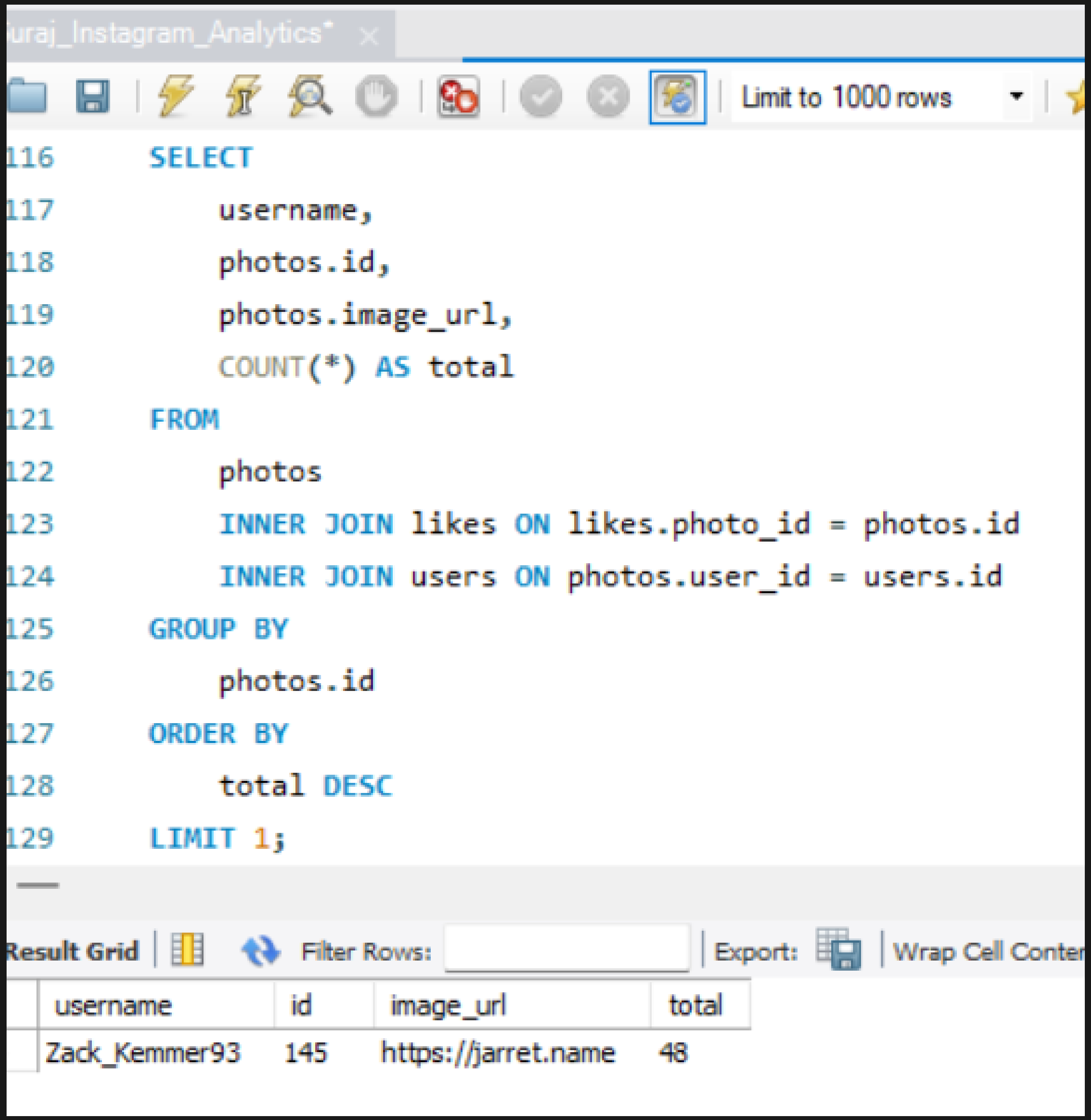
```
SELECT username
FROM users
WHERE id NOT IN (
SELECT DISTINCT user_id
FROM photos
);
```

username
Aniya_Hackett
Kasandra_Homenick
Jadyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

- we select only the "username" column from the "users" table in the main query
- Used WHERE clause to filters out users whose "id" values appear in the subquery.
- Selected all distinct "user_id" values from the "photos" table by using subquery.
- Used NOT IN operator in the main query to filters out any users whose "id" values appear in the subquery.
- And we got users who haven't uploaded any photos

Insights

Identify the winner of the contest and provide their details to the team



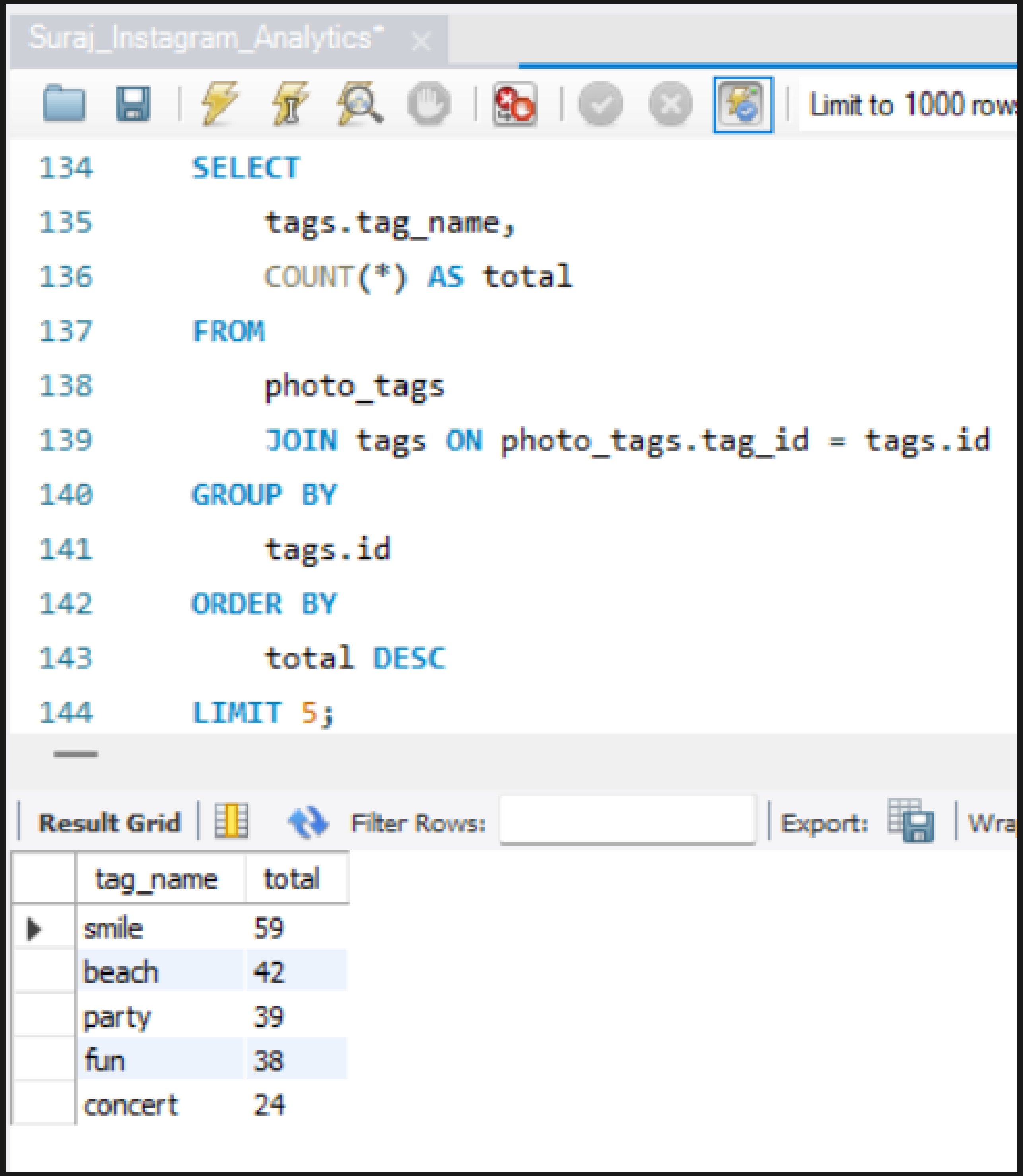
```
SELECT
    username,
    photos.id,
    photos.image_url,
    COUNT(*) AS total
FROM
    photos
    INNER JOIN likes ON likes.photo_id = photos.id
    INNER JOIN users ON photos.user_id = users.id
GROUP BY
    photos.id
ORDER BY
    total DESC
LIMIT 1;
```

username	id	image_url	total
Zack_Kemmer93	145	https://jarret.name	48

- we selects data from the "photos" table and joins it with the "likes" and "users" tables to match the photo, like, and user data.
- Then COUNT(*) function is used to count the number of likes for each photo.
- Further results are grouped by photo ID and sorted in descending order based on the number of likes.
- The LIMIT 1 used to show only the most-liked photo, along with its associated user information.

Insights

Identify and suggest the top 5 most commonly used hashtags on the platform



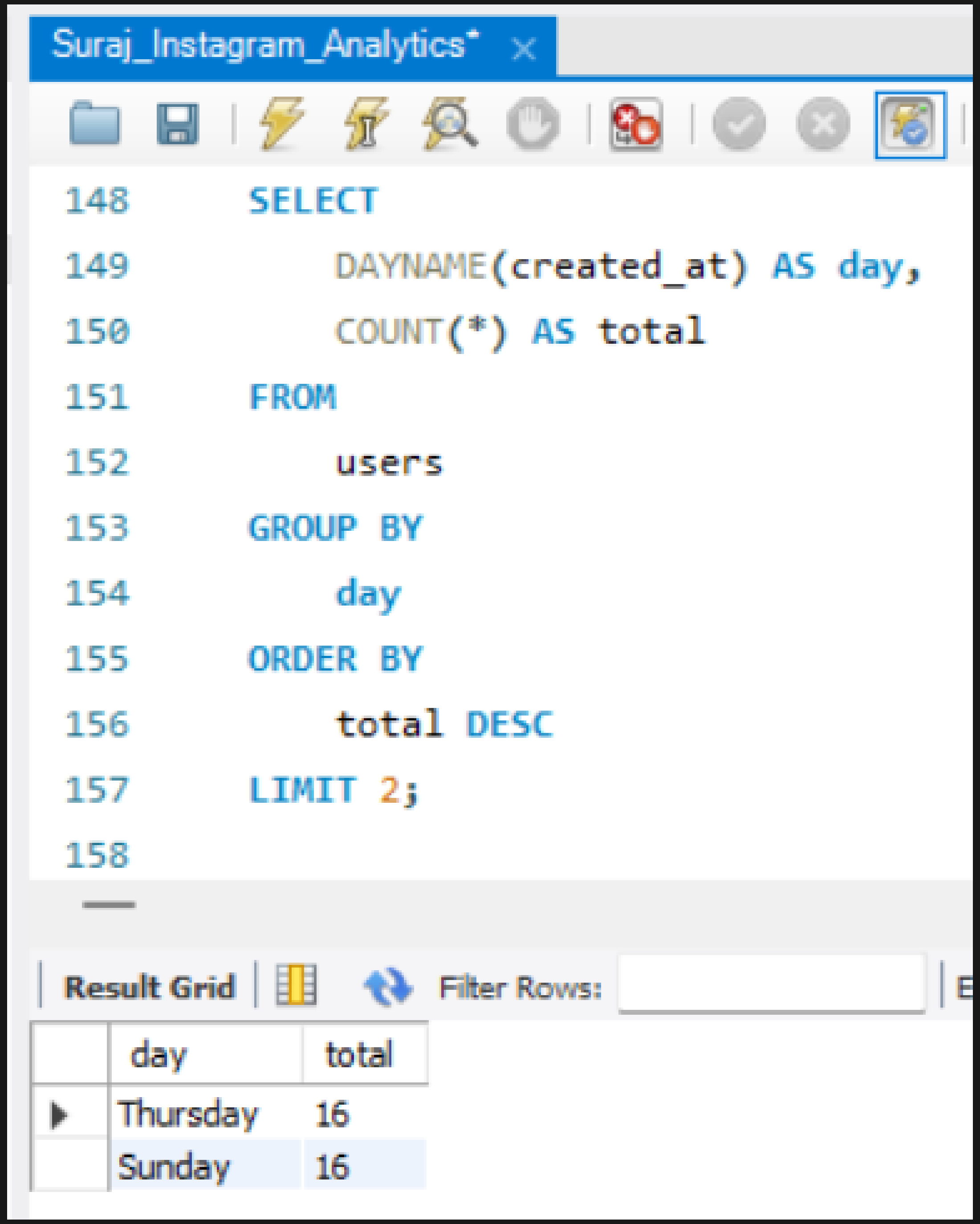
```sql
SELECT
    tags.tag_name,
    COUNT(*) AS total
FROM
    photo_tags
    JOIN tags ON photo_tags.tag_id = tags.id
GROUP BY
    tags.id
ORDER BY
    total DESC
LIMIT 5;
```

tag_name	total
smile	59
beach	42
party	39
fun	38
concert	24

- We selects data from the "photo_tags" table and joins it with the "tags" table to match tag IDs with their corresponding names.
- Then COUNT(*) function is used to count the number of times each tag appears in the "photo_tags" table.
- Further results are grouped by tag ID using the GROUP BY clause to aggregate the tag counts.
- Then ORDER BY clause sorts the results in descending order based on the total count of each tag.
- The LIMIT 5 clause limits the results to the top 5 most common tags, and the query get the tag name and total count for each of these tags.

Insights

What day of the week do most users register on? Provide insights on when to schedule an ad campaign



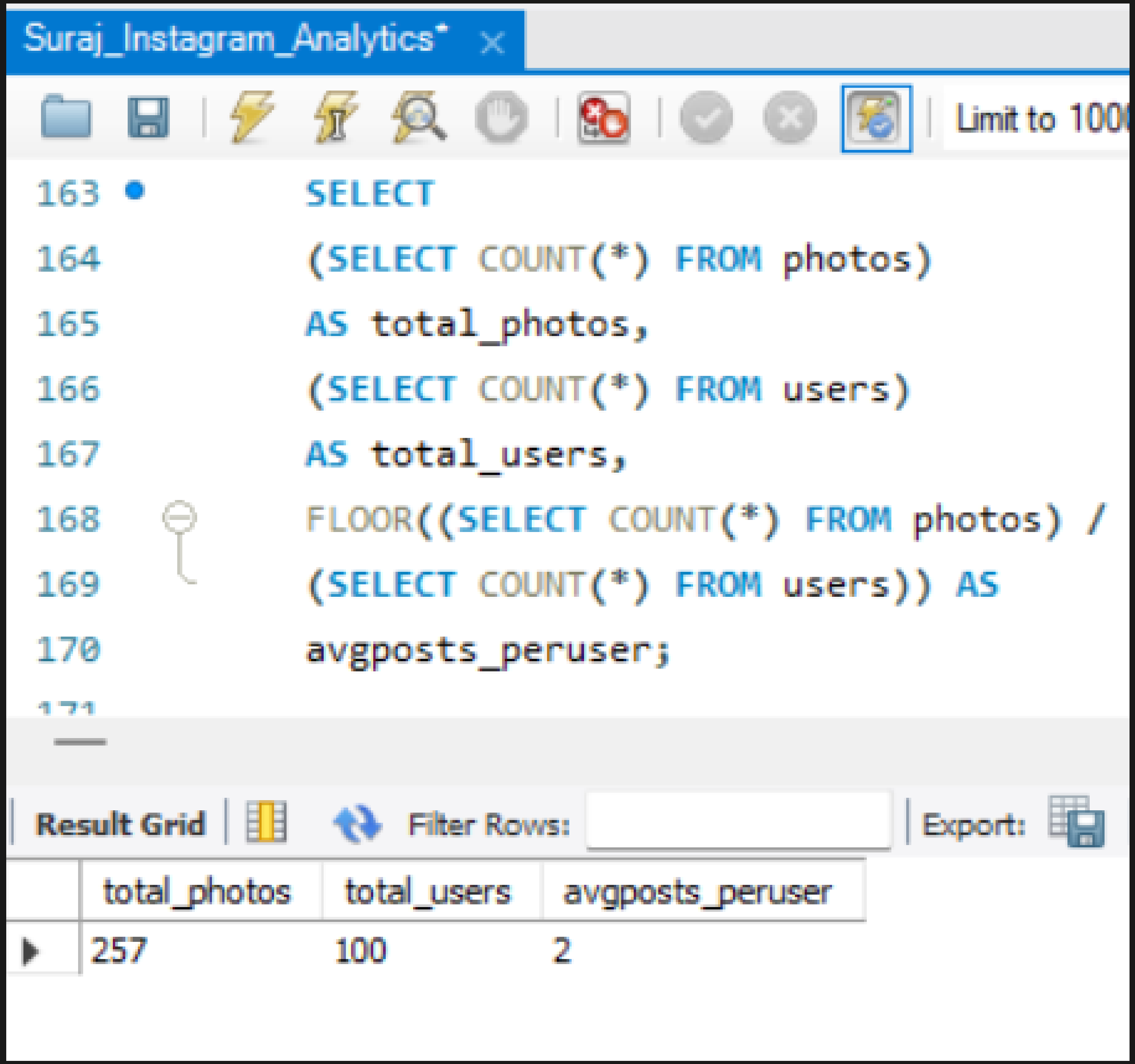
```
SELECT
    DAYNAME(created_at) AS day,
    COUNT(*) AS total
FROM
    users
GROUP BY
    day
ORDER BY
    total DESC
LIMIT 2;
```

```
day       total
Thursday  16
Sunday    16
```

- We have created_at column in from the users table by using DAYNAME function we can get days of account created
- by using COUNT(*) function we can get counts on each day of how many accounts get created
- Further used the ORDER BY clause to arrange in order where the highest account created to the lowest
- found 2 days where exact count of accounts generated
- Hence limit it by 2.

Insights

Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users



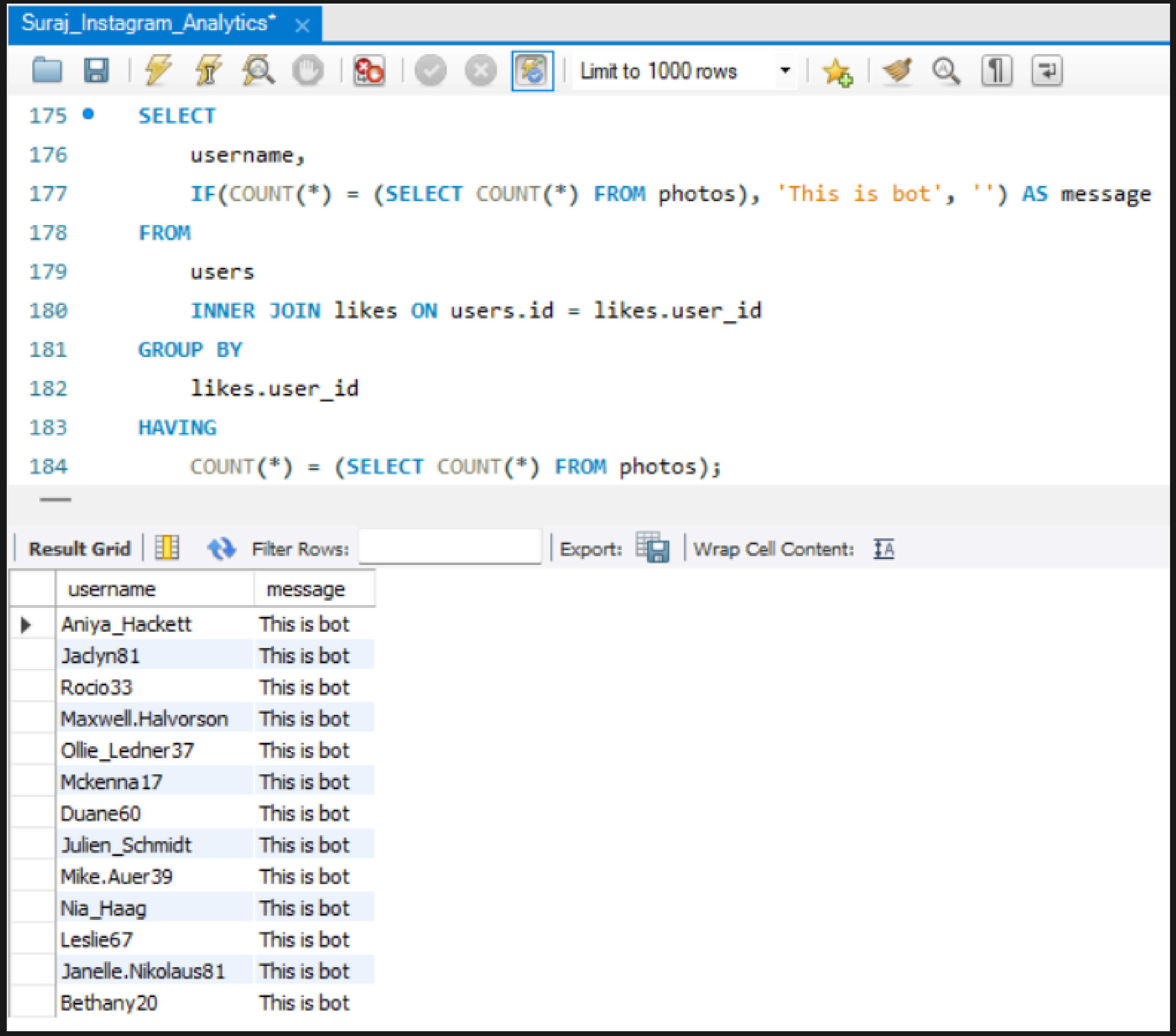
```
SELECT
(SELECT COUNT(*) FROM photos)
AS total_photos,
(SELECT COUNT(*) FROM users)
AS total_users,
FLOOR((SELECT COUNT(*) FROM photos) /
(SELECT COUNT(*) FROM users)) AS
avgposts_peruser;
```

```
  total_photos  total_users  avgposts_peruser
▶ 257           100          2
```

- First we count total no. of photos from "photos" table using COUNT(*) and assign as "total_photos"
- Then counts the total number of users in the "users" table using the COUNT(*) function and assigns the result as "total_users"
- Further we calculate average numbers of photos by didiving this two results and to round of we used "FLOOR" Function
- In the main query selects the three values and returns them as separate columns with the desired result.

Insights

Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).



```sql
SELECT
    username,
    IF(COUNT(*) = (SELECT COUNT(*) FROM photos), 'This is bot', '') AS message
FROM
    users
    INNER JOIN likes ON users.id = likes.user_id
GROUP BY
    likes.user_id
HAVING
    COUNT(*) = (SELECT COUNT(*) FROM photos);
```

username	message
Aniya_Hackett	This is bot
Jaclyn81	This is bot
Rocio33	This is bot
Maxwell.Halvorson	This is bot
Ollie_Ledner37	This is bot
Mckenna17	This is bot
Duane60	This is bot
Julien_Schmidt	This is bot
Mike.Auer39	This is bot
Nia_Haag	This is bot
Leslie67	This is bot
Janelle.Nikolaus81	This is bot
Bethany20	This is bot

- Bots are identified as they do not post anything but likes and comments on each and every photo
- So we focused on the user_id who has like every photo and count likes
- we also count the total photos to understand the total photos
- We filter out count matching with total photos from group by result of inner join likes on likes user id
- used if statement to display "this is bot" if total count of likes by user matches with total count of photos uploaded

Result

- While making this project helps to explore more SQL
- Also helps to start with My SQL database which is a great tool
- This project used data from users on Instagram which was comparatively smaller than it generates in actual
- But this help to the schema of the social media websites like Instagram
- it also gives the opportunity to explore various attributes like Data type, Nullabilty, Primary key, foreign key
- Explore SQL Functions like COUNT, MAX, DAYNAME, FLOOR, and IF
- While making subquery understand that there are many ways to do a single thing and I have more opportunities to explore it.

Attachment:- Link for project file

Thank You..