



INSTAGRAM USER ANALYSIS

SHRISTY PANDEY

PROJECT DESCRIPTION

01

Find out user's engagement and interaction with our product (software and mobile application) to drive business insight for marketing product and development team.

02

Find out the success of the app by measuring user engagement and improve the experience altogether while helping the business grow.

APPROACH

- Firstly we have to download the database file which is provided.
- Then the second approach should be how to use MYSQL to create that table and input values.
- Then we will work on how to create table.
- Before creating the table we have to gather some insight of database.
- After that we write the Queries to find the questions and answer.
- Then we analyse the information and make decision.
- The last step should be to make decision.

All the above mentioned point are help us to create an analysis report with better understanding. It will also easy our work to execute the plan.

Tech Stack Used

MYSQL server
8.0.28

MYSQL workbench
8.0.28

MS Excel 2020

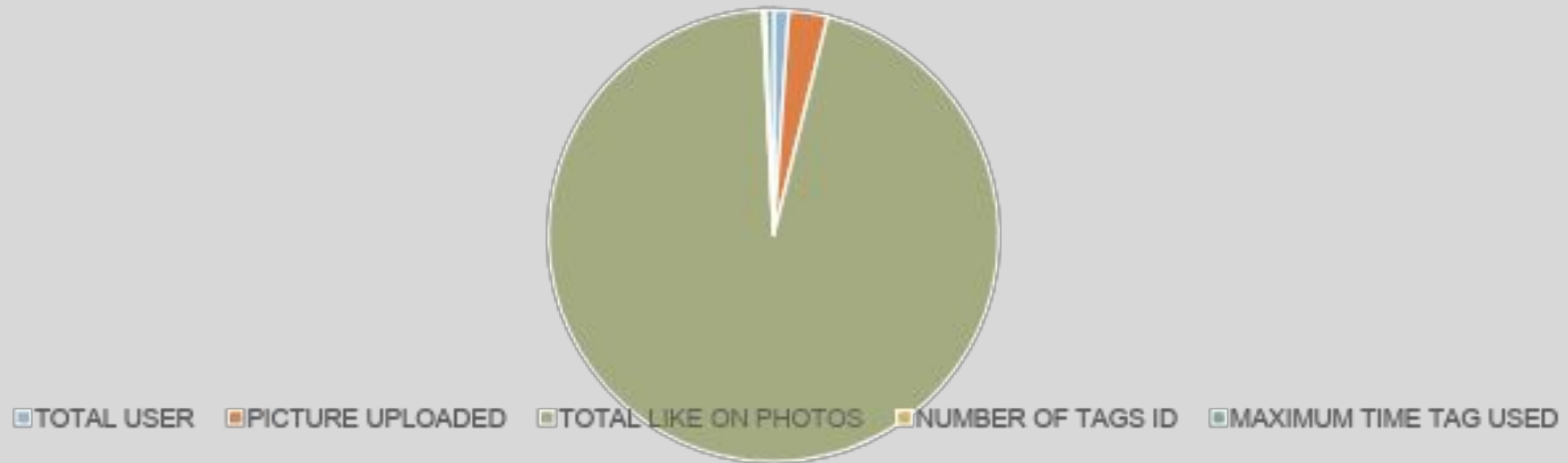
MS Word

Paint Application

INSIGHTS

- The main insight of this project is to find the users, total uploaded pictures on Instagram as well as like on photos,, number of tags, and how many times of tag is used. Here I have tried to show you some data by using pie chart.

INSTAGRAM USER INSIGHTS



Part A (Marketing)

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

```
SELECT * FROM users ORDER BY created_at LIMIT 5;
```

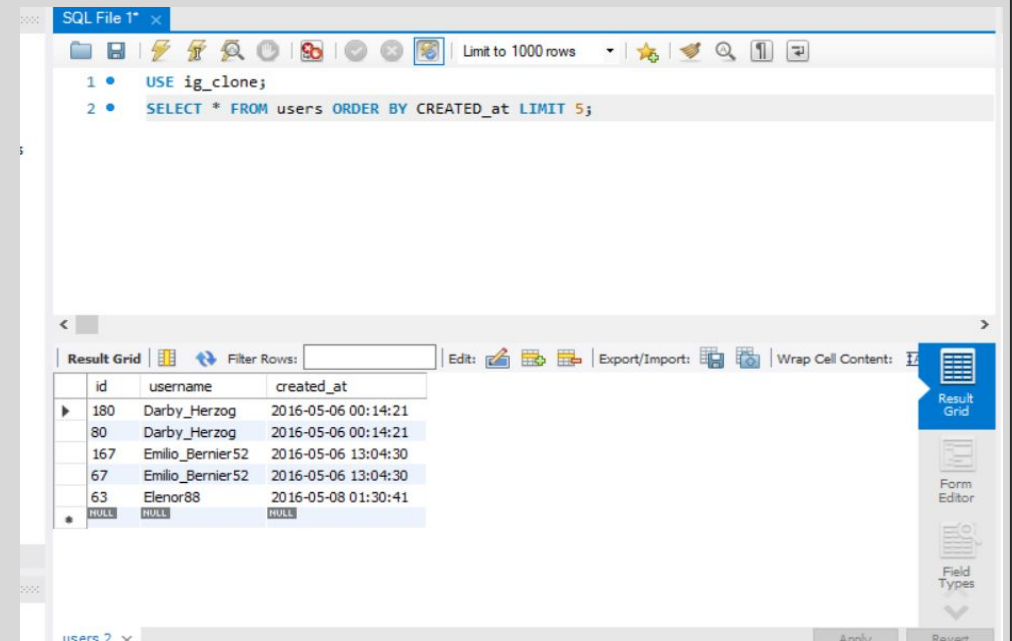
2. Find the users who have never posted a single photo on Instagram

```
SELECT I'd
```

```
Username
```

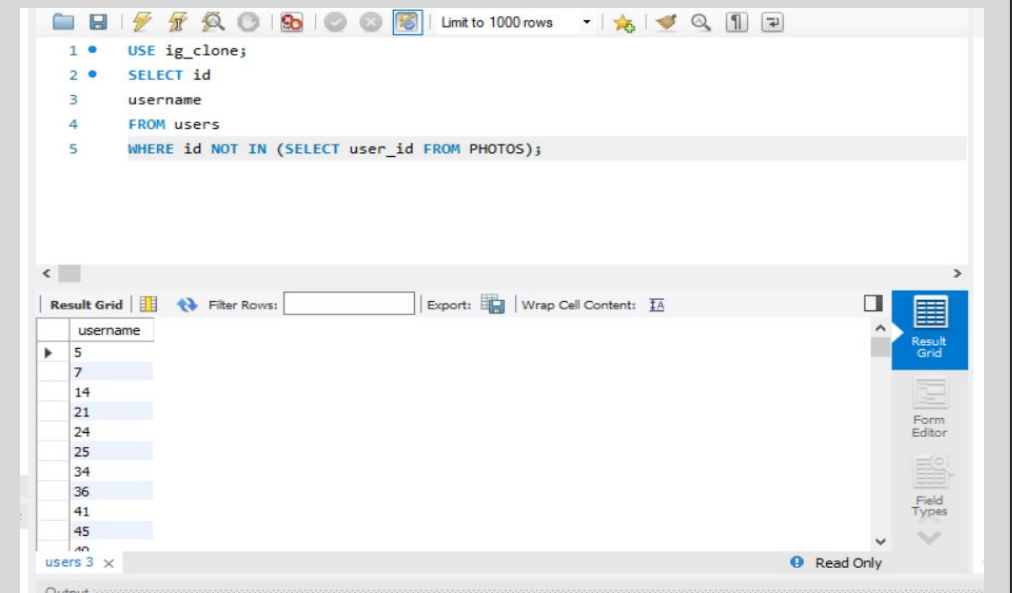
```
FROM users
```

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



The screenshot shows a SQL IDE window titled "SQL File 1*" with a toolbar at the top. The SQL editor contains two lines of code: `1 • USE ig_clone;` and `2 • SELECT * FROM users ORDER BY CREATED_at LIMIT 5;`. Below the editor, the "Result Grid" tab is active, displaying a table with three columns: `id`, `username`, and `created_at`. The table contains five rows of data, with the first row highlighted. The data is as follows:

id	username	created_at
180	Darby_Herzog	2016-05-06 00:14:21
80	Darby_Herzog	2016-05-06 00:14:21
167	Emilio_Bernier52	2016-05-06 13:04:30
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41



The screenshot shows a SQL IDE window titled "SQL File 1*" with a toolbar at the top. The SQL editor contains five lines of code: `1 • USE ig_clone;`, `2 • SELECT id`, `3 username`, `4 FROM users`, and `5 WHERE id NOT IN (SELECT user_id FROM PHOTOS);`. Below the editor, the "Result Grid" tab is active, displaying a table with one column: `username`. The table contains a list of usernames, with the first row highlighted. The data is as follows:

username
5
7
14
21
24
25
34
36
41
45

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

SELECT

users.username AS Name

likes.photo_id AS photo_id,

COUNT(likes.photo_id) AS phot_like_count

FROM users

INNER JOIN photos

ON users.id = photos.user_id

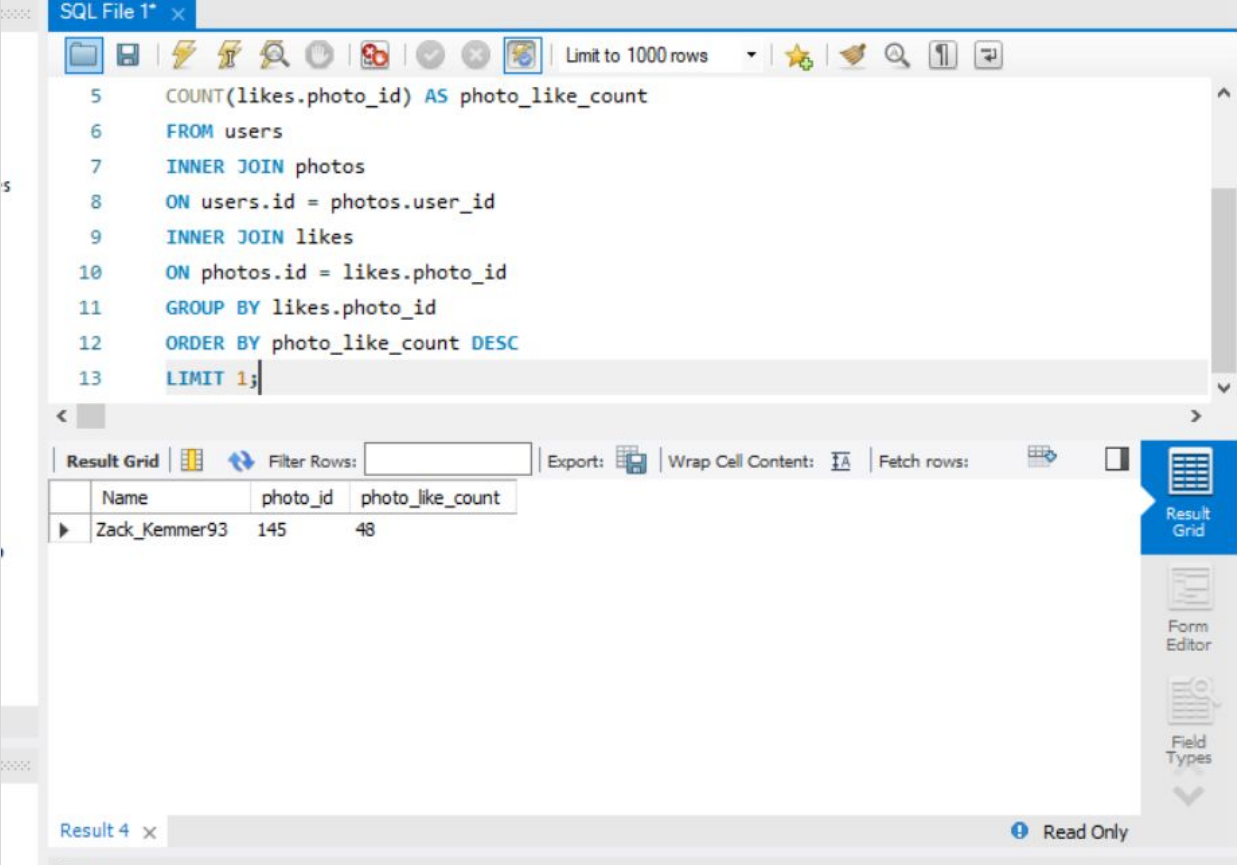
INNER JOIN likes

ON photos.id = likes.photo_id

GROUP BY likes.photo_id

ORDER BY photo_like_count DESC

LIMIT 1;



The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

```
5  COUNT(likes.photo_id) AS photo_like_count
6  FROM users
7  INNER JOIN photos
8  ON users.id = photos.user_id
9  INNER JOIN likes
10 ON photos.id = likes.photo_id
11 GROUP BY likes.photo_id
12 ORDER BY photo_like_count DESC
13 LIMIT 1;
```

Below the query editor, the "Result Grid" is displayed, showing a single row of results:

Name	photo_id	photo_like_count
Zack_Kemmer93	145	48

The IDE interface includes a toolbar with various icons, a "Filter Rows" input field, and buttons for "Export", "Wrap Cell Content", "Fetch rows", "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 4 x" and "Read Only".

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

SELECT

tags.id,

COUNT(tag_name) AS tag_count

FROM tags

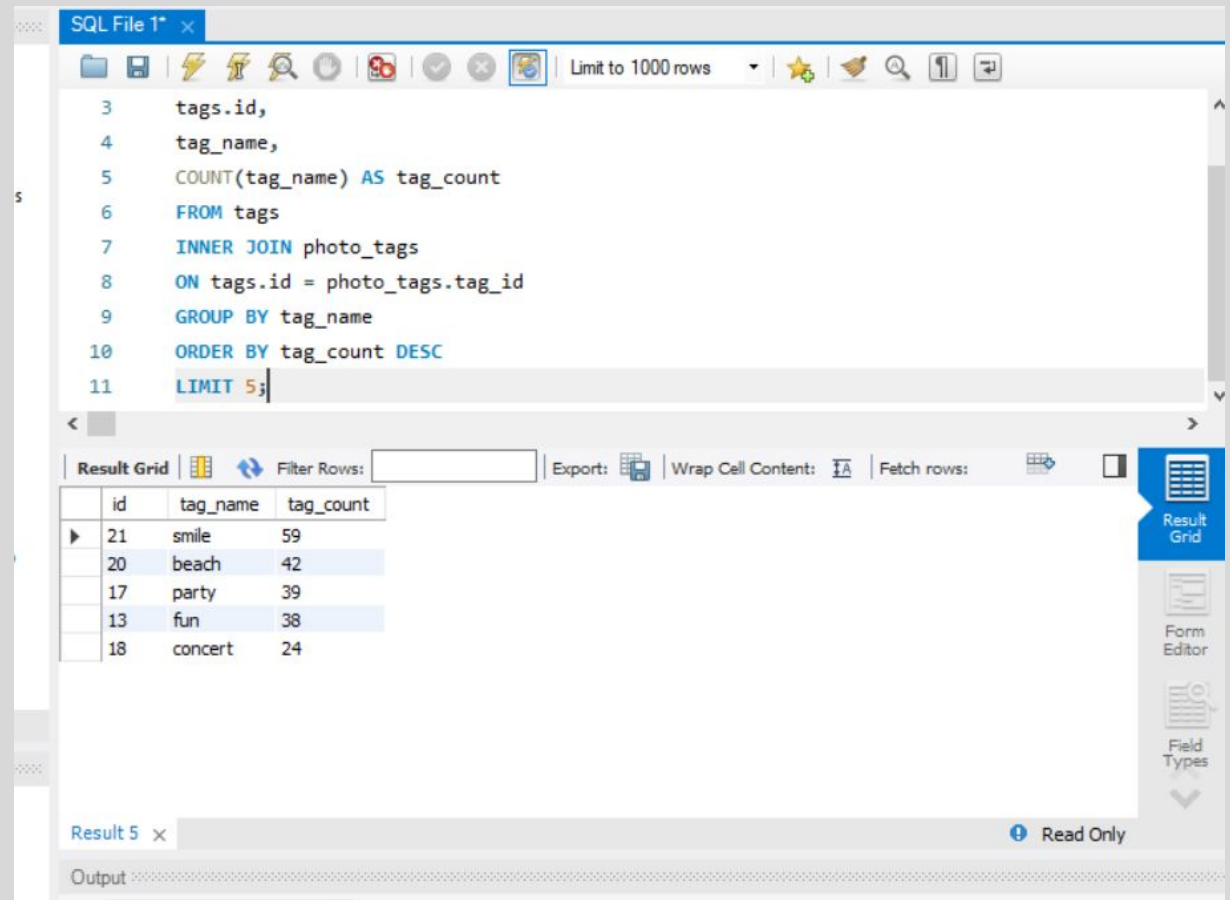
INNER JOIN photo_tags

ON tags.id = photo_tags.id

GROUP BY tag_name

ORDER BY tag_count DESC

LIMIT 5;



The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

```
3 tags.id,  
4 tag_name,  
5 COUNT(tag_name) AS tag_count  
6 FROM tags  
7 INNER JOIN photo_tags  
8 ON tags.id = photo_tags.tag_id  
9 GROUP BY tag_name  
10 ORDER BY tag_count DESC  
11 LIMIT 5;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are shown in a table with columns: id, tag_name, and tag_count. The table contains 5 rows of data:

	id	tag_name	tag_count
▶	21	smile	59
	20	beach	42
	17	party	39
	13	fun	38
	18	concert	24

The IDE interface includes various toolbars at the top and right, and a status bar at the bottom indicating "Result 5 x" and "Read Only".

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

SELECT

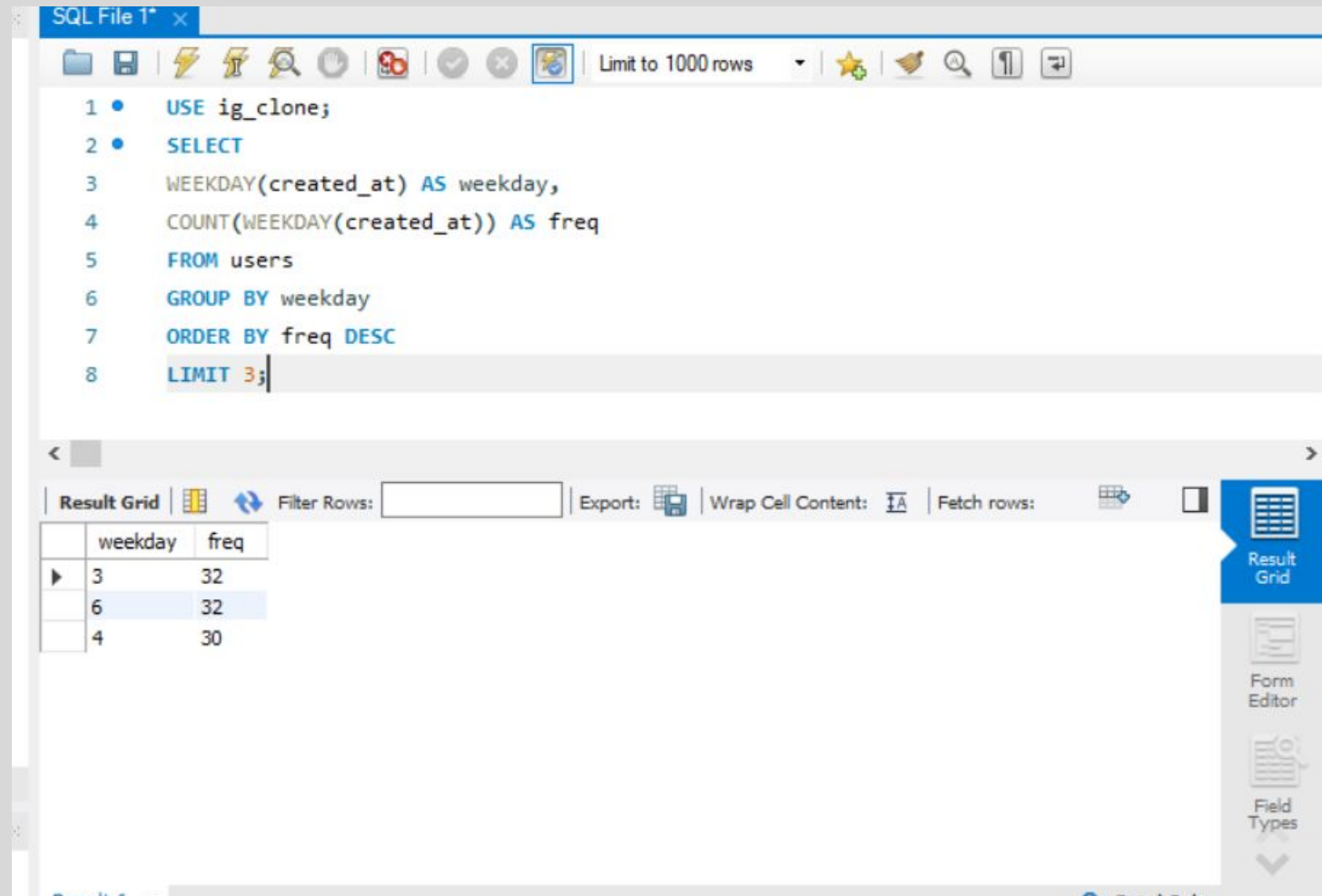
WEEKDAY (created_at) AS freq

FROM users

GROUP BY weekday

ORDER BY freq DESC

LIMIT 3;



The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

```
1 • USE ig_clone;  
2 • SELECT  
3   WEEKDAY(created_at) AS weekday,  
4   COUNT(WEEKDAY(created_at)) AS freq  
5 FROM users  
6 GROUP BY weekday  
7 ORDER BY freq DESC  
8 LIMIT 3;
```

The results pane below the query editor displays a table with two columns: "weekday" and "freq". The table contains three rows of data, with the second row (weekday 6, freq 32) highlighted in blue. The interface includes a toolbar with various icons and a sidebar on the right with buttons for "Result Grid", "Form Editor", and "Field Types".

	weekday	freq
▶	3	32
	6	32
	4	30

Part B (Investor Metrics)

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

With t1 as

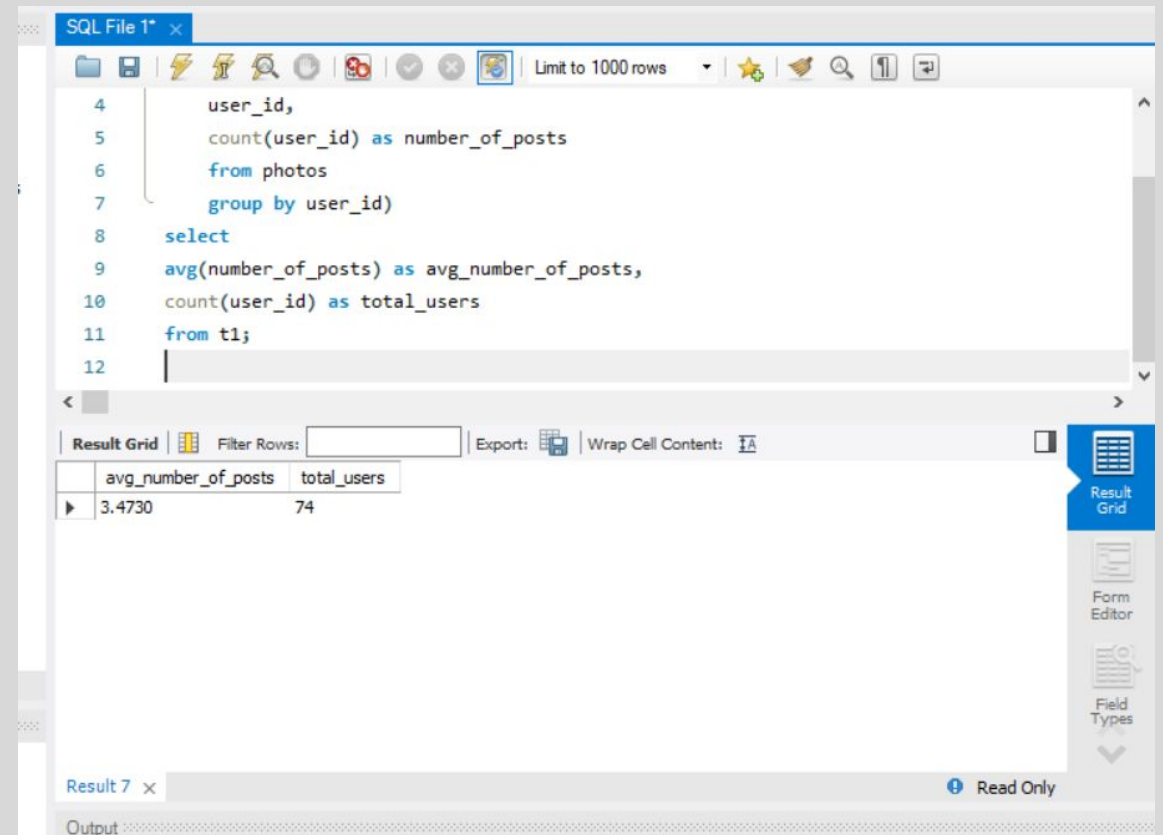
```
(select  
    user_id,  
    count(user_id) as number_of_posts  
from photos  
group by (users_id)
```

select

Avg(number_of_posts) as avg_number_of_posts,

Count(user_id) as total_users

From t1;



The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

```
4      user_id,  
5      count(user_id) as number_of_posts  
6  from photos  
7  group by user_id)  
8  select  
9  avg(number_of_posts) as avg_number_of_posts,  
10 count(user_id) as total_users  
11 from t1;  
12
```

Below the query editor, the "Result Grid" is displayed, showing the following data:

avg_number_of_posts	total_users
3.4730	74

The IDE interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a sidebar with buttons for "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 7 x" and "Read Only".

2. 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).

SELECT

Likes.user_id,

COUNT(DISTINCT photo_id) AS number_of_photos_liked

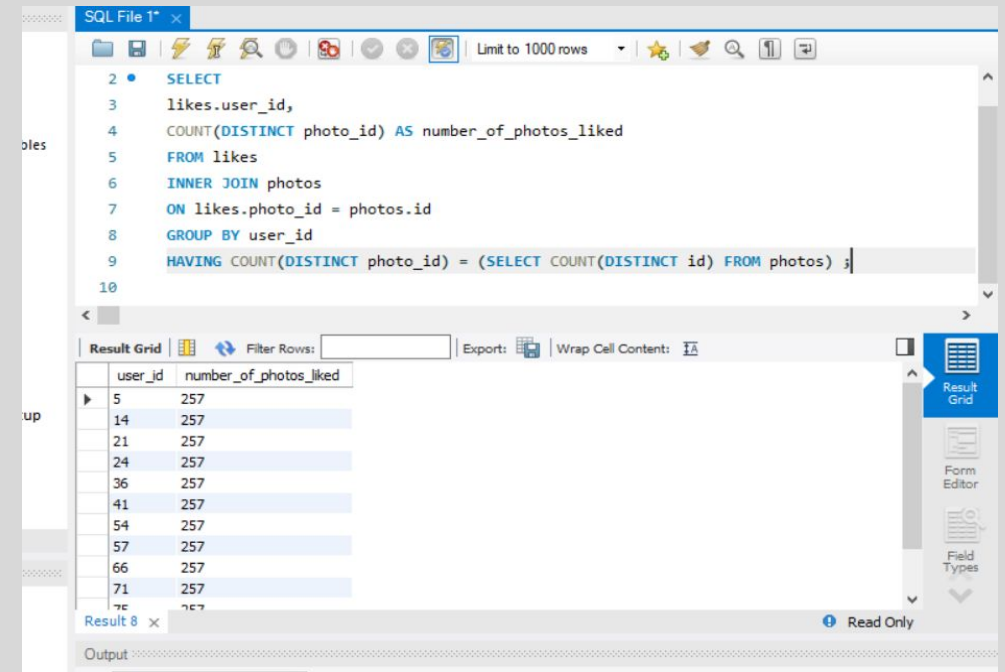
FROM likes

INNER JOIN photos

ON like.photo_id = photo.id

GROUP BY user_id

HAVING COUNT(DISTINCT photo_id) = (SELECT COUNT(DISTINCT id) FROM photos) ;



The screenshot shows a SQL IDE window titled "SQL File 1*" with a query editor and a result grid. The query is as follows:

```
2 SELECT
3   likes.user_id,
4   COUNT(DISTINCT photo_id) AS number_of_photos_liked
5 FROM likes
6 INNER JOIN photos
7   ON likes.photo_id = photos.id
8 GROUP BY user_id
9 HAVING COUNT(DISTINCT photo_id) = (SELECT COUNT(DISTINCT id) FROM photos) ;
10
```

The result grid displays the following data:

user_id	number_of_photos_liked
5	257
14	257
21	257
24	257
36	257
41	257
54	257
57	257
66	257
71	257
76	257

The IDE interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The result grid also has a "Filter Rows" input, an "Export" button, and a "Wrap Cell Content" checkbox. The status bar at the bottom indicates "Result 8" and "Read Only".

RESULT

- Rewards oldest users for there activities.
- Work on more personalized and interactive feature.
- Held more n more campaigns for more users activation.
- Also we can send promotional email for the 1st photo post on insta, who never uploaded any post.
- Make more smile related positive tags.
- Campaigns held on Sunday and Thursday.

