INSTAGRAM USER PROBLEMS WITH THEIR QUERIES AND RESULT

Q1-> Loyal User Reward: The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

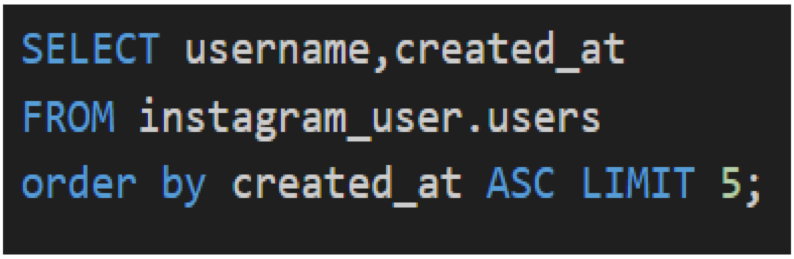
-Your Task: Identify the five oldest users on Instagram from the provided database.

SOLUTION->

Step-by-Step Approach:

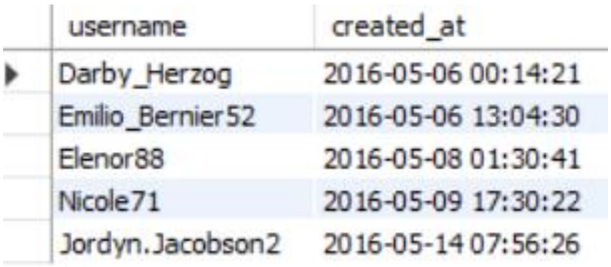
1. Understand the Data:
    - The table instagram_user.users contains user account information.
    - Relevant columns:
        - username: the Instagram handle of each user.
        - created_at: the date and time when the user registered on the platform.
2. Sort by Registration Date:
    - To find the oldest users, we need to sort the records by the created_at column in ascending order (ASC) — so the earliest dates come first.
3. Limit the Output:
    - Since the task is to find only the top 5 oldest users, we apply the LIMIT 5 clause to return just five records from the sorted list.

QUERY



```
SELECT username,created_at
FROM instagram_user.users
order by created_at ASC LIMIT 5;
```

RESULT



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

QUESTION 2->Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.

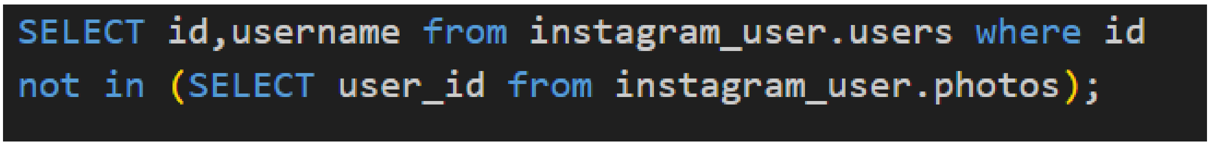
Your Task: Identify users who have never posted a single photo on Instagram.

SOLUTION->

STEP 1 -> SELECT id, username FROM instagram_user.users:
Retrieves the user ID and username for all users.

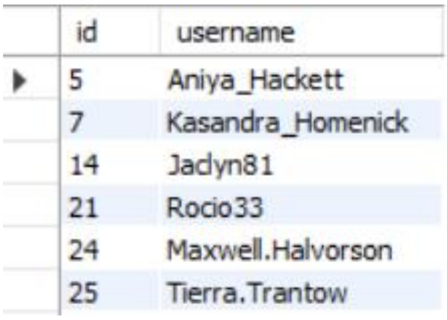
STEP 2-> WHERE id NOT IN (...):
Filters out users whose IDs are found in the list of user_ids in the photos table.

STEP 3-> SELECT user_id FROM instagram_user.photos:
Returns all user IDs that have posted at least one photo.



```
SELECT id,username from instagram_user.users where id
not in (SELECT user_id from instagram_user.photos);
```

RESULT ->



id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow

QUESTION 3-> Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.

-- Your Task: Determine the winner of the contest and provide their details to the team

SOLUTION->

Step-by-Step Procedure:

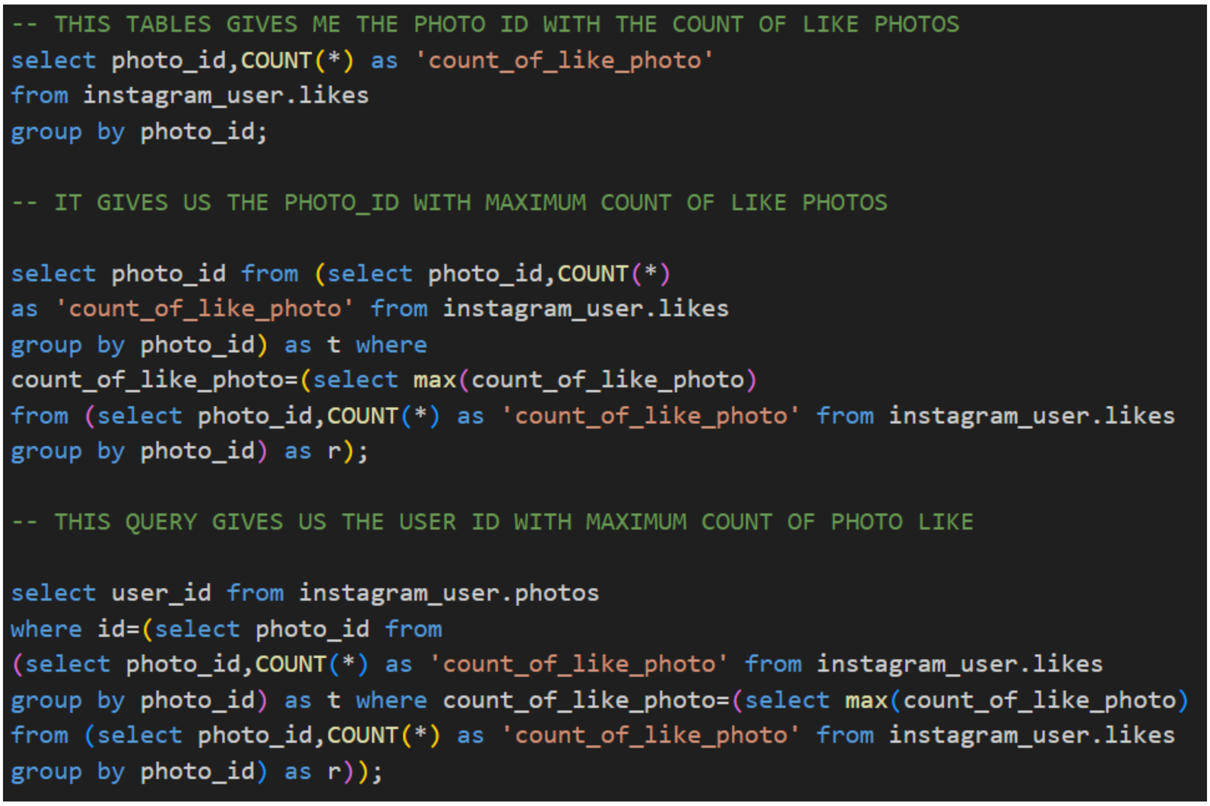
Step 1: Count Likes per Photo

Step 2: Find the Photo with the Maximum Likes

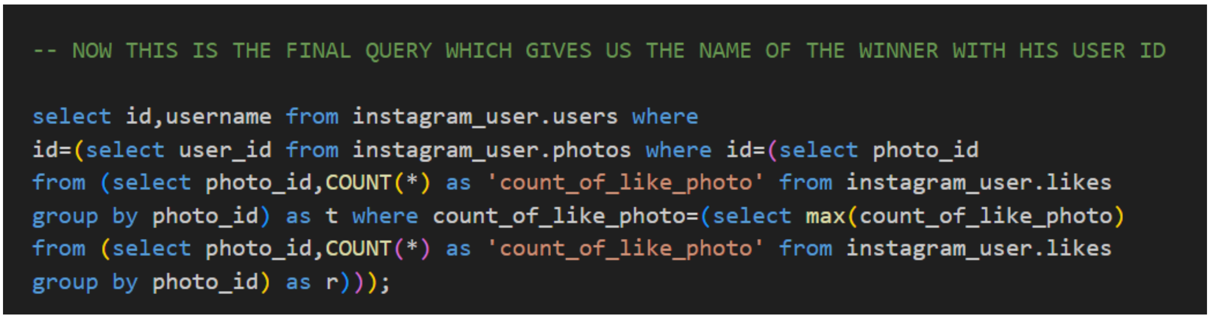
Step 3: Get the User Who Posted That Photo

Step 4: Get the User's Details (Winner)

QUERY

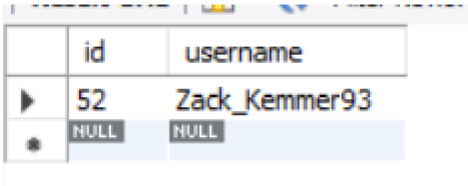


```
-- THIS TABLES GIVES ME THE PHOTO ID WITH THE COUNT OF LIKE PHOTOS
select photo_id,COUNT(*) as 'count_of_like_photo'
from instagram_user.likes
group by photo_id;

-- IT GIVES US THE PHOTO_ID WITH MAXIMUM COUNT OF LIKE PHOTOS

select photo_id from (select photo_id,COUNT(*)
as 'count_of_like_photo' from instagram_user.likes
group by photo_id) as t where
count_of_like_photo=(select max(count_of_like_photo)
from (select photo_id,COUNT(*) as 'count_of_like_photo' from instagram_user.likes
group by photo_id) as r);

-- THIS QUERY GIVES US THE USER ID WITH MAXIMUM COUNT OF PHOTO LIKE

select user_id from instagram_user.photos
where id=(select photo_id from
(select photo_id,COUNT(*) as 'count_of_like_photo' from instagram_user.likes
group by photo_id) as t where count_of_like_photo=(select max(count_of_like_photo)
from (select photo_id,COUNT(*) as 'count_of_like_photo' from instagram_user.likes
group by photo_id) as r));
```



```
-- NOW THIS IS THE FINAL QUERY WHICH GIVES US THE NAME OF THE WINNER WITH HIS USER ID

select id,username from instagram_user.users where
id=(select user_id from instagram_user.photos where id=(select photo_id
from (select photo_id,COUNT(*) as 'count_of_like_photo' from instagram_user.likes
group by photo_id) as t where count_of_like_photo=(select max(count_of_like_photo)
from (select photo_id,COUNT(*) as 'count_of_like_photo' from instagram_user.likes
group by photo_id) as r)));
```

RESULT



id	username
52	Zack_Kemmer93
NULL	NULL

QUESTION 4-> A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

-- Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

SOLUTION->

Step 1: Understand the Data Structure

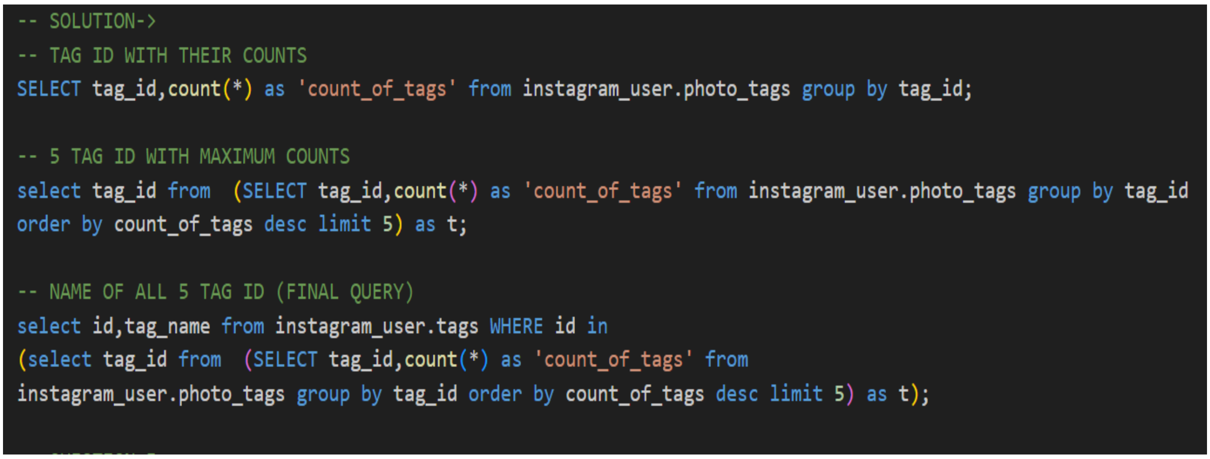
- photo_tags: A mapping table that connects photo_id with tag_id.
    - Column used: tag_id
- tags: Contains tag details.
    - Columns used: id, tag_name

Step 2: Count How Often Each Tag is Used

Step 3: Find the Top 5 Most Frequently Used Tag IDs

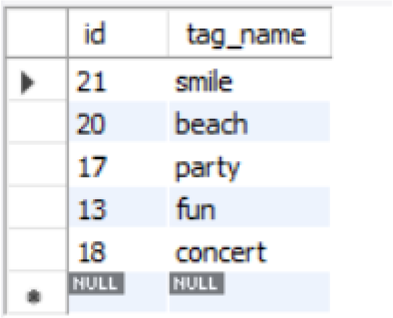
Step 4: Get the Tag Names for These Top 5 Tags

QUERY



```
-- SOLUTION->
-- TAG ID WITH THEIR COUNTS
SELECT tag_id,count(*) as 'count_of_tags' from instagram_user.photo_tags group by tag_id;

-- 5 TAG ID WITH MAXIMUM COUNTS
select tag_id from  (SELECT tag_id,count(*) as 'count_of_tags' from instagram_user.photo_tags group by tag_id
order by count_of_tags desc limit 5) as t;

-- NAME OF ALL 5 TAG ID (FINAL QUERY)
select id,tag_name from instagram_user.tags WHERE id in
(select tag_id from  (SELECT tag_id,count(*) as 'count_of_tags' from
instagram_user.photo_tags group by tag_id order by count_of_tags desc limit 5) as t);
```

RESULT



id	tag_name
21	smile
20	beach
17	party
13	fun
18	concert
NULL	NULL

QUESTION 5->Ad Campaign Launch: The team wants to know the best day of the week to launch ads.

Your Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

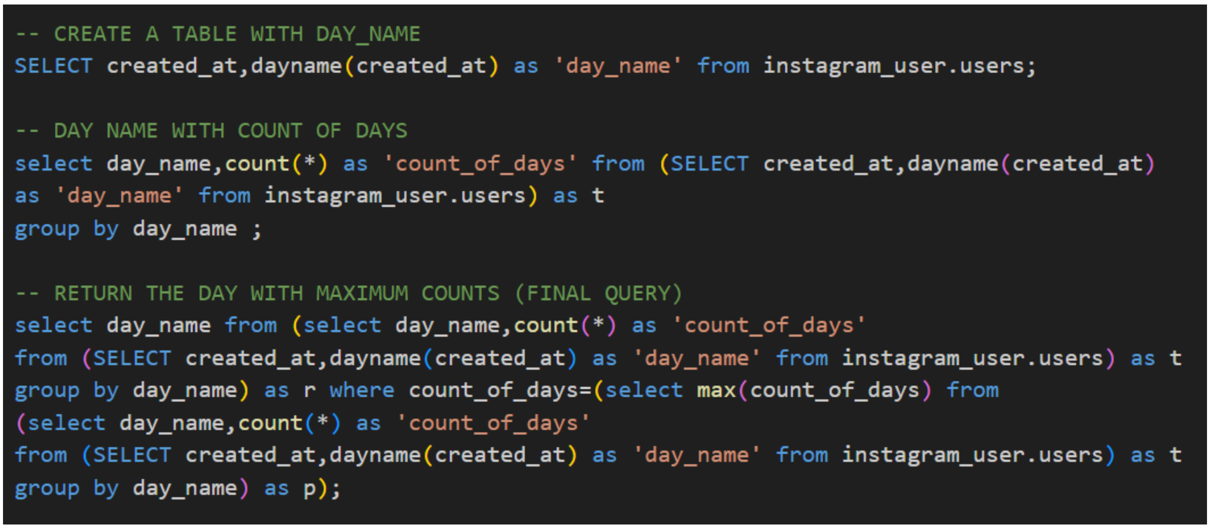
SOLUTION->

STEP -1. Extracting Day Names from User Creation Dates

STEP-2. Counting Users by Day of the Week

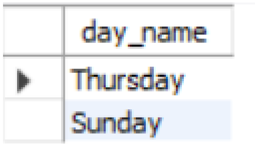
STEP-3. Identifying the Day with the Maximum User Creations

QUERY



```
-- CREATE A TABLE WITH DAY_NAME
SELECT created_at,dayname(created_at) as 'day_name' from instagram_user.users;

-- DAY NAME WITH COUNT OF DAYS
select day_name,count(*) as 'count_of_days' from (SELECT created_at,dayname(created_at)
as 'day_name' from instagram_user.users) as t
group by day_name ;

-- RETURN THE DAY WITH MAXIMUM COUNTS (FINAL QUERY)
select day_name from (select day_name,count(*) as 'count_of_days'
from (SELECT created_at,dayname(created_at) as 'day_name' from instagram_user.users) as t
group by day_name) as r where count_of_days=(select max(count_of_days) from
(select day_name,count(*) as 'count_of_days'
from (SELECT created_at,dayname(created_at) as 'day_name' from instagram_user.users) as t
group by day_name) as p);
```

RESULT



day_name
Thursday
Sunday

QUESTION 6 -> Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

Your Task: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

SOLUTION->

STEP 1.Retrieve Each User's Post Count

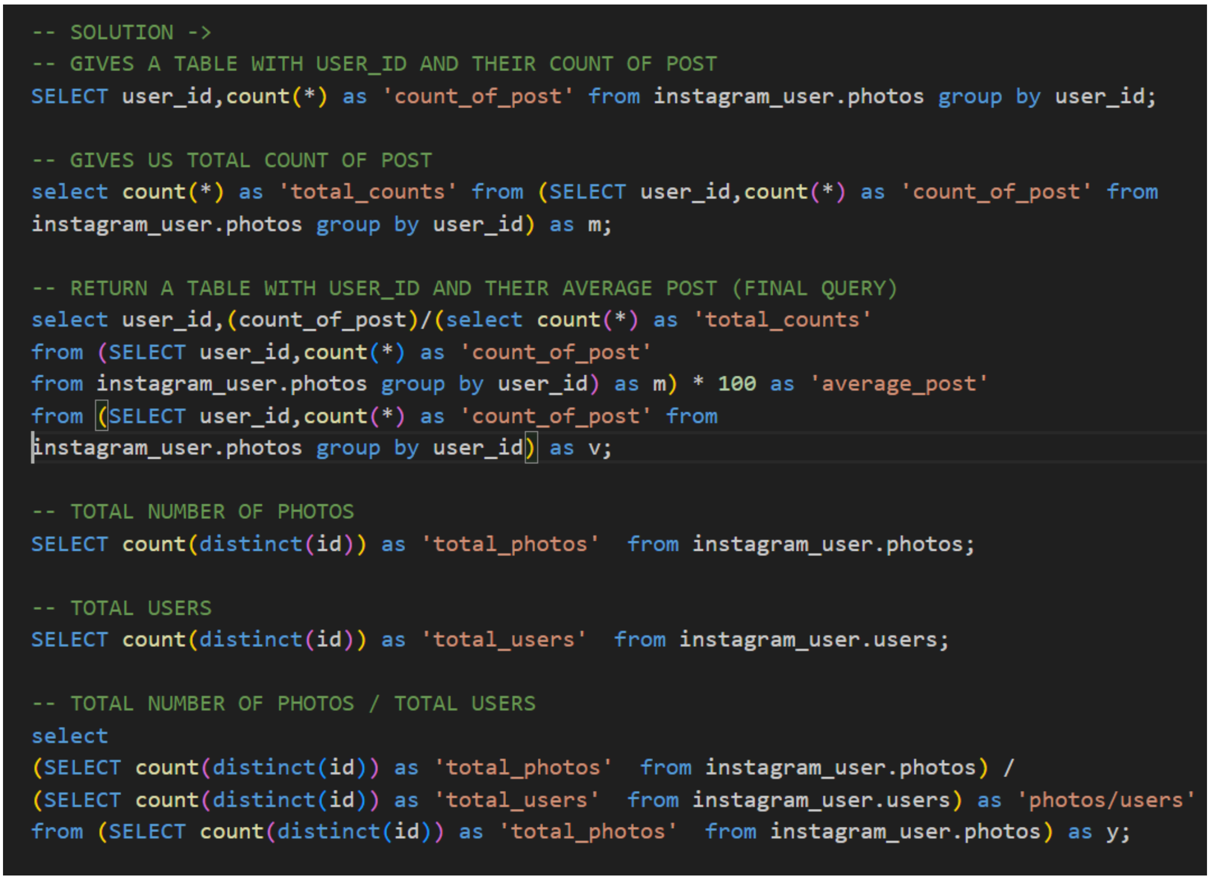
STEP 2. Calculate Total Number of Users Who Posted

STEP 3. Compute Each User's Contribution to Total Posts (Percentage)

STEP 4. Determine Total Number of Photos

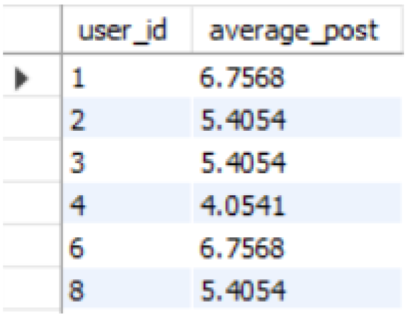
STEP 5. Determine Total Number of Users

QUERY

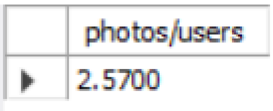


```
-- SOLUTION ->
-- GIVES A TABLE WITH USER_ID AND THEIR COUNT OF POST
SELECT user_id,count(*) as 'count_of_post' from instagram_user.photos group by user_id;

-- GIVES US TOTAL COUNT OF POST
select count(*) as 'total_counts' from (SELECT user_id,count(*) as 'count_of_post' from
instagram_user.photos group by user_id) as m;

-- RETURN A TABLE WITH USER_ID AND THEIR AVERAGE POST (FINAL QUERY)
select user_id,(count_of_post)/(select count(*) as 'total_counts'
from (SELECT user_id,count(*) as 'count_of_post'
from instagram_user.photos group by user_id) as m) * 100 as 'average_post'
from (SELECT user_id,count(*) as 'count_of_post' from
instagram_user.photos group by user_id) as v;

-- TOTAL NUMBER OF PHOTOS
SELECT count(distinct(id)) as 'total_photos'  from instagram_user.photos;

-- TOTAL USERS
SELECT count(distinct(id)) as 'total_users'  from instagram_user.users;

-- TOTAL NUMBER OF PHOTOS / TOTAL USERS
select
(SELECT count(distinct(id)) as 'total_photos'  from instagram_user.photos) /
(SELECT count(distinct(id)) as 'total_users'  from instagram_user.users) as 'photos/users'
from (SELECT count(distinct(id)) as 'total_photos'  from instagram_user.photos) as y;
```

RESULT



	user_id	average_post
▶	1	6.7568
	2	5.4054
	3	5.4054
	4	4.0541
	6	6.7568
	8	5.4054



	photos/users
▶	2.5700

QUESTION 7->Investors want to know if the platform is crowded with fake and dummy accounts.

Your Task: Identify users (potential bots) who have liked every single photo on the site as this is not typically possible for a normal user.

SOLUTION->

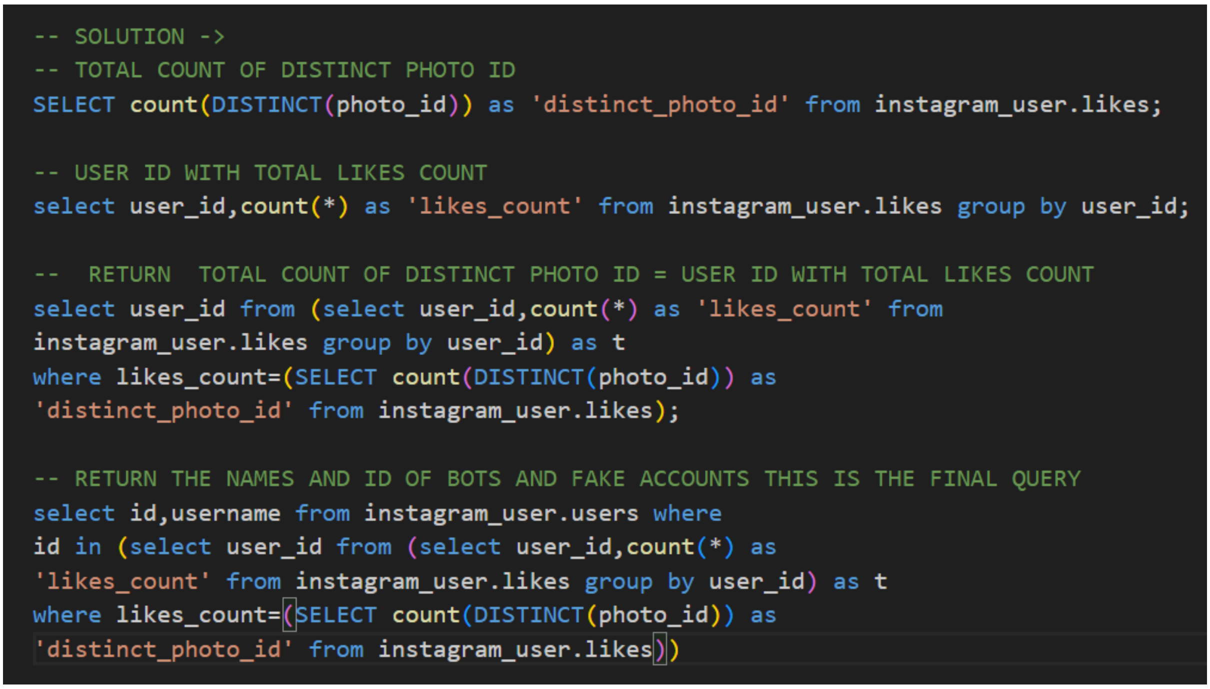
STEP 1-> Calculate Total Number of Unique Photos Liked

STEP2->Determine Total Likes Made by Each User

STEP 3->Identify Users Who Have Liked All Unique Photos

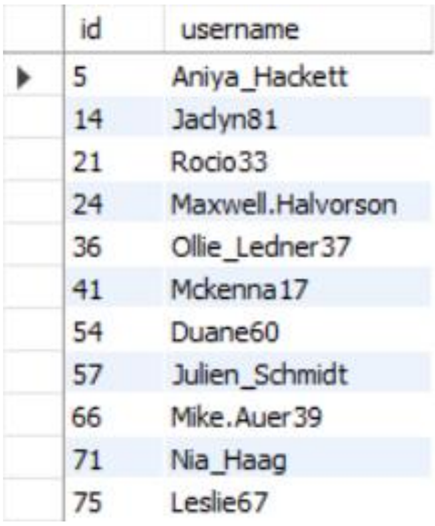
STEP 4->Retrieve Usernames and IDs of Potential Bot Accounts

QUERY



```
-- SOLUTION ->
-- TOTAL COUNT OF DISTINCT PHOTO ID
SELECT count(DISTINCT(photo_id)) as 'distinct_photo_id' from instagram_user.likes;

-- USER ID WITH TOTAL LIKES COUNT
select user_id,count(*) as 'likes_count' from instagram_user.likes group by user_id;

--  RETURN  TOTAL COUNT OF DISTINCT PHOTO ID = USER ID WITH TOTAL LIKES COUNT
select user_id from (select user_id,count(*) as 'likes_count' from
instagram_user.likes group by user_id) as t
where likes_count=(SELECT count(DISTINCT(photo_id)) as
'distinct_photo_id' from instagram_user.likes);

-- RETURN THE NAMES AND ID OF BOTS AND FAKE ACCOUNTS THIS IS THE FINAL QUERY
select id,username from instagram_user.users where
id in (select user_id from (select user_id,count(*) as
'likes_count' from instagram_user.likes group by user_id) as t
where likes_count=(SELECT count(DISTINCT(photo_id)) as
'distinct_photo_id' from instagram_user.likes))
```

RESULT



id	username
5	Aniya_Hackett
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
36	Ollie_Ledner37
41	Mckenna17
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
71	Nia_Haag
75	Leslie67