1. Identify the five oldest users on Instagram from the provided database.

Query:

```
select id, username, created_at as Period from users
order by created_at asc
limit 5;
```

Output:

Result Grid | Filter Rows:

id	username	Period
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

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

Query:

```
SELECT u.id, u.username
FROM users u
LEFT JOIN photos p ON u.id = p.user_id
WHERE p.user_id IS NULL;
```

Output:

Result Grid | Filter Rows:

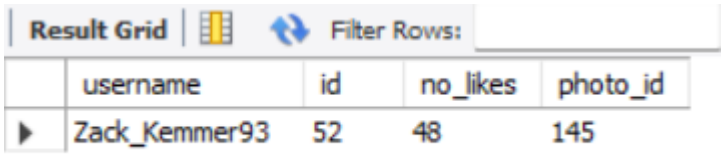
id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jadyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow

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

Query:

```
select u.username, u.id, count(l.photo_id) as no_likes, l.photo_id from users u
join photos p
on u.id = p.user_id
join likes l
on p.id = l.photo_id
group by l.photo_id
order by no_likes desc
limit 1;
```

Output:



Result Grid | Filter Rows:

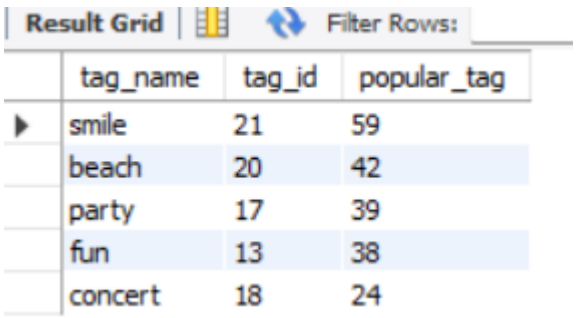
username	id	no_likes	photo_id
Zack_Kemmer93	52	48	145

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

Query:

```
select t.tag_name, pt.tag_id, count(pt.tag_id) as popular_tag from tags t
join photo_tags pt
on t.id = pt.tag_id
group by pt.tag_id
order by popular_tag desc
limit 5;
```

Output:



Result Grid | Filter Rows:

tag_name	tag_id	popular_tag
smile	21	59
beach	20	42
party	17	39
fun	13	38
concert	18	24

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

Query:

```
select dayname(created_at) as day_of_week, count(dayname(created_at)) as num_registrations
from users
group by day_of_week
order by num_registrations DESC
LIMIT 1;
```

Output:

Result Grid | Filter Rows:

day_of_week	num_registrations
Thursday	16

6. 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.

Query:

```
select avg(photo_count) as avg_post_per_user
from (select u.id as user_id, u.username, count(p.id) as photo_count from users u
join photos p
on u.id = p.user_id
group by u.id) user_photo_count;
```

Output:

Result Grid | Filter

avg_post_per_user
3.4730

Query:

```
select
(select count(id) from photos)/(select count(id) from users) as avg_photos_per_users;
```

Output:

Result Grid | Filter R

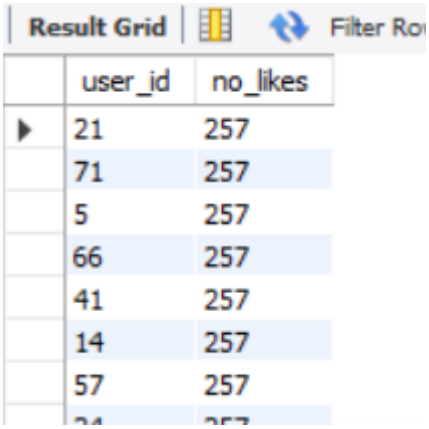
avg_photos_per_users
2.5700

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

Query:

```
select user_id, count(photo_id) as no_likes from likes
group by user_id
order by no_likes desc;
```

Output:



user_id	no_likes
21	257
71	257
5	257
66	257
41	257
14	257
57	257