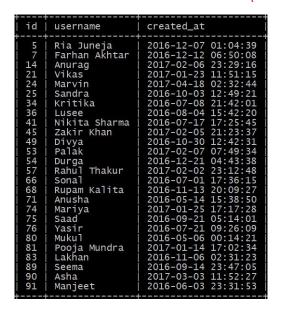
Q1) We want to reward our users who have been around the longest. Find the 5 oldest users.

SELECT \* FROM users ORDER BY created at LIMIT 5;

id	username	created_at
80	   Mukul	2016-05-06 00:14:21
67	Soujanya	2016-05-06 13:04:30
63	Shivam Sid	2016-05-08 01:30:41
95	Afreen	2016-05-09 17:30:22
38	Nandita	2016-05-14 07:56:26

- Q2) What day of the week do most users register on? We need to figure out when to schedule and ad campaign.
  - SELECT DAYNAME(created\_at) as day, COUNT(created\_at) as day\_count FROM users GROUP BY day ORDER BY day\_count DESC LIMIT 1;

- Q3) We want to target our inactive users with an email campaign. Find the users who have never posted a photo.
  - SELECT \* FROM users WHERE id NOT IN (SELECT user\_id FROM photos);



Q4) We're running a new contest to see who can get the most likes on a single photo. WHO WON?

SELECT users.id as user\_id, users.username as username, photos.id as photo\_id, photos.image\_url as url, COUNT(likes.user\_id) as like\_count FROM likes JOIN photos ON likes.photo\_id=photos.id JOIN users ON photos.user\_id=users.id GROUP BY likes.photo\_id ORDER BY like\_count DESC LIMIT 1;

user_id	username	photo_id	url	like_count
52	Nisha	145	https://jarret.name	48

Q5) Our Investors want to know. How many times does the average user post? Total number of photos/total number of users.

SELECT ROUND((SELECT COUNT(id) FROM photos)/(SELECT COUNT(id) FROM users),2) as photo\_posting\_avg\_time;

## Q6) User ranking by postings higher to lower?

SELECT users.id, users.username, COUNT(photos.id) as photos\_posted FROM users JOIN photos ON users.id=photos.user\_id GROUP BY users.id ORDER BY photos\_posted DESC;



- Q7) Total numbers of users who have posted at least one time.
  - SELECT COUNT(DISTINCT user\_id) as user\_posted\_atleast\_once FROM photos;

- Q8) A brand wants to know which hashtags to use in a post. What are the top 5 most commonly used hashtags?
  - SELECT tags.id as tag\_id, tags.tag\_name as tag\_name, COUNT(photo\_tags.tag\_id) as tag\_count FROM tags JOIN photo\_tags ON tags.id=photo\_tags.tag\_id GROUP BY tags.id ORDER BY tag\_count DESC LIMIT 5;

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

- Q9) We have a small problem with bots on our site. Find users who have liked every single photo on the site and Commented on every single photo.
  - SELECT users.id as user\_id, users.username as username, COUNT(users.id) as total\_likes\_by\_user FROM users JOIN likes ON users.id=likes.user\_id GROUP BY users.id HAVING total\_likes\_by\_user = (SELECT COUNT(\*) FROM photos);

user_id	username	total_likes_by_user
5 14 21 24 36 41 54 57 66 71 75 76 91	Ria Juneja Anurag Vikas Marvin Lusee Nikita Sharma Durga Rahul Thakur Sonal Anusha Saad Yasir Manjeet	257 257 257 257 257 257 257 257 257 257

SELECT users.id as user\_id, users.username as username, COUNT(comments.user\_id) as total\_comment FROM users JOIN comments ON users.id=comments.user\_id GROUP BY users.id HAVING total\_comment = (SELECT COUNT(DISTINCT comments.photo\_id) FROM comments);

user_id	username	total_comment
5 14 21 24 36 41 54 57 66 71 75 76 91	Ria Juneja Anurag Vikas Marvin Lusee Nikita Sharma Durga Rahul Thakur Sonal Anusha Saad Yasir Manjeet	257 257 257 257 257 257 257 257 257 257

Q10) We also have a problem with celebrities. Find users who have never commented on a photo.

SELECT users.id as user\_id, users.username as username, comments.comment\_text FROM users LEFT JOIN comments ON users.id=comments.user\_id WHERE comments.comment\_text IS NULL;

user_id	username	comment_text
1	Sachin	NULL
7	Farhan Akhtar	NULL
23	Sanjeet	NULL
25	Sandra	NULL
29	Mashad	NULL
34	Kritika	NULL
45	Zakir Khan	NULL
49	Divya	NULL
51	Simran	NULL
53	Palak	NULL
58	Riyanshi Gupta	NULL
59	Sandeep	NULL
64	Sireesha	NULL
68	Rupam Kalita	NULL
74	Mariya	NULL
77	Vaibhav	NULL
80	Muku1	NULL
81	Pooja Mundra	NULL
83	Lakhan	NULL
86	Vaishnavi	NULL
88	Laxmi	NULL
89	Seema	NULL
90	Asha	NULL