```
USE 10kcoders;
CREATE TABLE Insta_post (
    post_id INT PRIMARY KEY,
    user_id INT,
    username VARCHAR(50),
    caption VARCHAR(255),
    likes INT,
    comments INT,
    created_at DATE
);

INSERT INTO Insta_post VALUES
(101, 1, 'yash_123', 'Morning vibes 🌞', 120, 15, '2023-01-01'),
(102, 2, 'neha_insta', 'Travel diaries ✈️', 300, 40, '2023-02-10'),
(103, 3, 'raj_cool', 'My new painting 🎨', 95, 10, '2023-02-20'),
(104, 1, 'yash_123', 'Happy Holi! 🌈', 500, 80, '2023-03-08'),
(105, 4, 'anita_star', 'Workout motivation 💪', 250, 25, '2023-03-15'),
(106, 5, 'vikas_travel', 'Beautiful sunset 🌅', 400, 60, '2023-03-20'),
(107, 3, 'raj_cool', 'Foodie life 🍕', 150, 20, '2023-04-05'),
(108, 2, 'neha_insta', 'Best friends ❤️', 600, 100, '2023-04-12'),
(109, 5, 'vikas_travel', 'Mountain trek 🏔️', 550, 90, '2023-04-25'),
(110, 4, 'anita_star', 'Self care Sunday 🧘‍♀️', 200, 18, '2023-05-01');

select * from insta_post;
-- questions:

-- Find posts with likes greater than the average likes of all posts.

select * from Insta_post where likes >(select avg(likes) from Insta_post);

-- Find posts that have more comments than the average comments of all posts.
select * from Insta_post where comments >(
select avg(comments) from Insta_post);

-- Find the post(s) with the maximum likes.
select * from Insta_post where likes =(
select max(likes) from Insta_post);

-- Find the post(s) with the minimum likes.

select * from Insta_post where likes =(
select min(likes) from Insta_post);

-- Find the caption of the second most liked post.
select caption from Insta_post where caption =(
select * from Insta_post order by likes desc limit 1 offset 1)
;
```

```
-- Find posts created on the same date as the most liked post.
select * from Insta_post where created_at =(
select created_at from Insta_post where likes =(
select max(likes) from Insta_post));

-- Find posts with likes greater than twice the likes of post_id = 103.
select * from Insta_post where likes>(
select 2*likes from Insta_post where post_id = 103);

-- Find posts made before the earliest post date.

select * from insta_post;
select * from insta_post where created_at <(select min(created_at) from insta_post);

-- Find posts made after the latest post date.

select * from insta_post where created_at >(select max(created_at) from insta_post);


-- Find posts that have comments equal to the maximum comments.

select * from insta_post where comments = (select max(comments) from insta_post);

-- Find posts that have likes equal to the average likes.

select * from insta_post where likes= (select avg(likes) from insta_post);

-- Find posts that have likes greater than the average likes in March 2023.
select * from insta_post where likes >(
select avg(likes) from insta_post where created_at between "2023-03-01"and "2023-03-31"
);

-- Find posts with comments greater than the average comments in April 2023.
select * from insta_post where comments >(
select avg(comments) from insta_post where created_at between "2023-04-01" and
"2023-04-30"
);
-- Find posts with likes equal to the maximum likes in April 2023.

select * from insta_post where likes =(
select max(likes) from insta_post where created_at between "2023-04-01" and "2023-04-30"
);

-- Find posts of the user who posted the most liked post.
select * from insta_post where likes =(
select max(likes) from insta_post);

-- Find posts of the user who posted the least liked post.
```

```
select * from insta_post where likes =(
select min(likes) from insta_post);

-- Find posts where caption length is greater than the average caption length.
select * from insta_post where length(caption) >(
select avg(length(caption)) from insta_post);

-- Find posts created by users who posted at least one post with more than 500 likes.

select * from insta_post where username in(
select username from insta_post where likes >500);

-- Find posts created by users who posted at least one post with fewer than 100 likes.
select * from insta_post where username =(
select username from insta_post where likes<100
);
-- Find posts where likes are greater than all posts of 'raj_cool'.

select * from insta_post where likes >(
select sum(likes) from insta_post where username ="raj_cool"
);
-- Find posts where likes are greater than any post of 'yash_123'.
select username,count(username) from insta_post group by username;
select * from insta_post where username in(
select username from insta_post group by username having count(post_id)>(
select count(post_id) from insta_post where username = "yash_123"));

-- Find posts where comments are higher than the average comments of 'neha_insta'.

select * from insta_post where comments>(
select avg(comments) from insta_post where username ="neha_insta"
);
-- Find posts created after the earliest post of 'vikas_travel'.
select * from insta_post where created_at >(
select min(created_at) from insta_post where username = "vikas_travel"
);
-- Find posts created before the latest post of 'anita_star'.
select * from insta_post where created_at<(
select min(created_at) from insta_post where username = "anita_star"
);
-- Find posts of users who posted more times than 'yash_123'.
select * from insta_post where username =(
select username from insta_post group by username having count(*)>(
select count(*) from insta_post where username ="yash_123"));

-- Find posts of users who posted fewer times than 'raj_cool'.

select * from insta_post where username =(
```

```
select username from insta_post group by username having count(*)>(
select count(*) from insta_post where username ="raj_cool"));

-- Find posts that have likes greater than the overall average likes plus 100.
select * from insta_post where likes >(
select avg(likes)+100 from insta_post);

-- Find posts that have comments greater than the overall average comments plus 20.
select * from insta_post where comments >(
select avg(comments)+20 from insta_post);

-- Find posts where likes are equal to the median likes of all posts.
-- median is based on no of obs and here n is even then median is n/2 term i,e; 5th term
after arranging in descending order
select * from insta_post where likes =(
select likes from insta_post order by likes desc limit 1 offset 4
);
 -- when n is odd
-- SELECT *FROM insta_postWHERE likes = (SELECT likes FROM insta_post ORDER BY
likes DESC
--     LIMIT 1 OFFSET (((SELECT COUNT(*) FROM insta_post) + 1) / 2) - 1
-- );

-- Find posts that were created on the same date as another post.
select * from insta_post where created_at in(
select created_at from insta_post group by created_at having count(*)>1);
```