A) Marketing Analysis:

1. Loyal User Reward:

SELECT username, created\_at

FROM users

ORDER BY created\_at ASC

LIMIT 5;

2. Inactive User Engagement:

SELECT username

FROM users

LEFT JOIN photos ON users.id = photos.user\_id

WHERE photos.id IS NULL;

3. Contest Winner Declaration:

SELECT users.username, photos.image\_url, COUNT(likes.photo\_id) as like\_count

FROM photos

JOIN users ON photos.user\_id = users.id

LEFT JOIN likes ON photos.id = likes.photo\_id

GROUP BY likes.photo\_id, users.username, photos.image\_url

ORDER BY like\_count DESC

LIMIT 1;

4. Hashtag Research:

SELECT tags.tag\_name, COUNT(photo\_tags.tag\_id) as tag\_count

FROM tags

JOIN photo\_tags ON tags.id = photo\_tags.tag\_id

GROUP BY photo\_tags.tag\_id

ORDER BY tag\_count DESC

LIMIT 5;

5. Ad Campaign Launch:

SELECT DAYNAME(created\_at) as registration\_day, COUNT(id) as user\_count

FROM users

GROUP BY registration\_day

ORDER BY user\_count DESC

LIMIT 1;

B) Investor Metrics:

1. User Engagement:

-- Average posts per user

SELECT AVG(post\_count) as avg\_posts\_per\_user

FROM (

SELECT user\_id, COUNT(id) as post\_count

FROM photos

GROUP BY user\_id

) as user\_posts;

-- Total photos to users ratio

SELECT (SELECT COUNT(\*) FROM photos) / (SELECT COUNT(\*) FROM users) as photos\_to\_users\_ratio;

2. Bots & Fake Accounts:

SELECT users.username

FROM users

JOIN likes ON users.id = likes.user\_id

GROUP BY likes.user\_id

HAVING COUNT(likes.photo\_id) = (SELECT COUNT(\*) FROM photos);