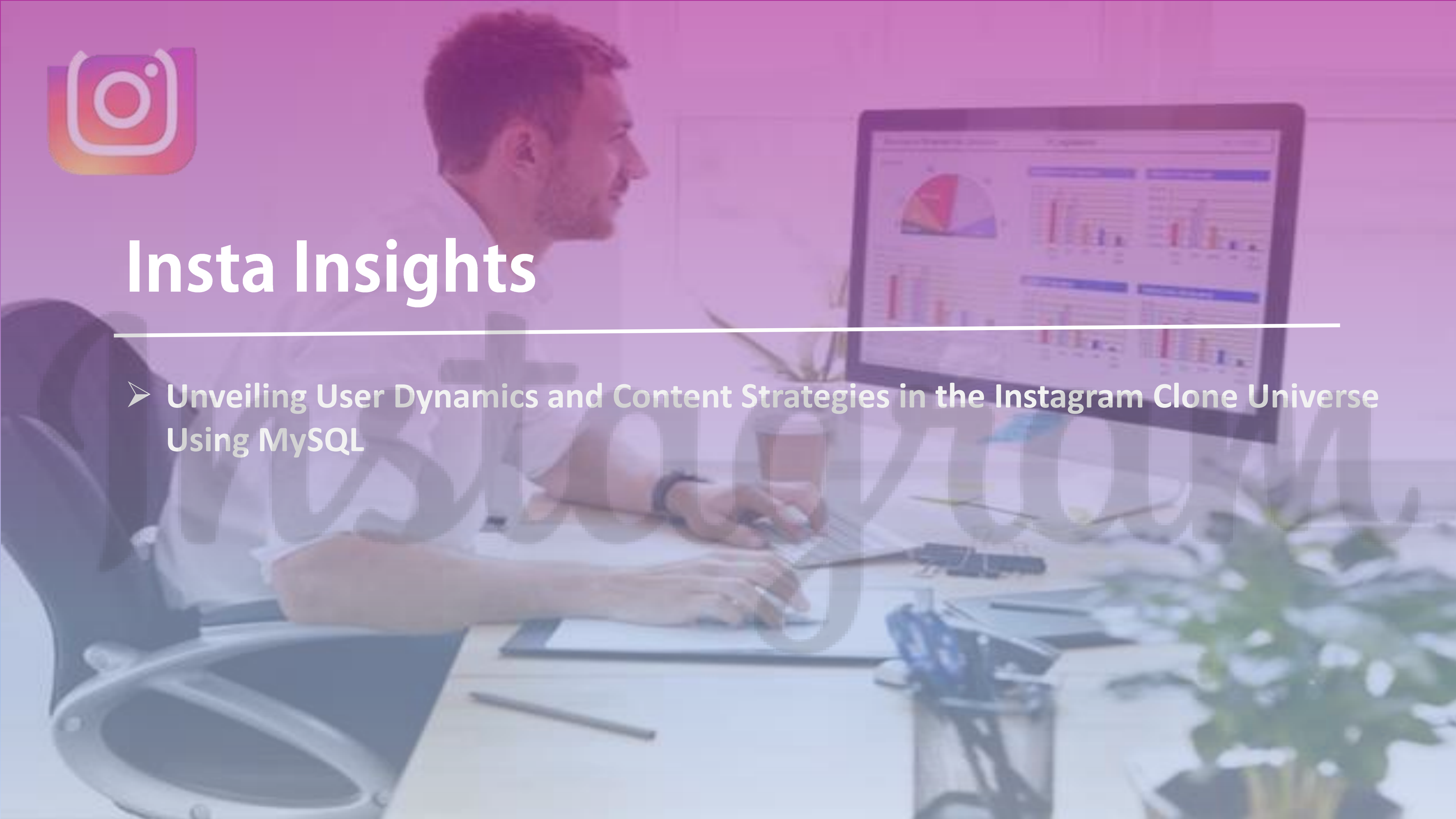


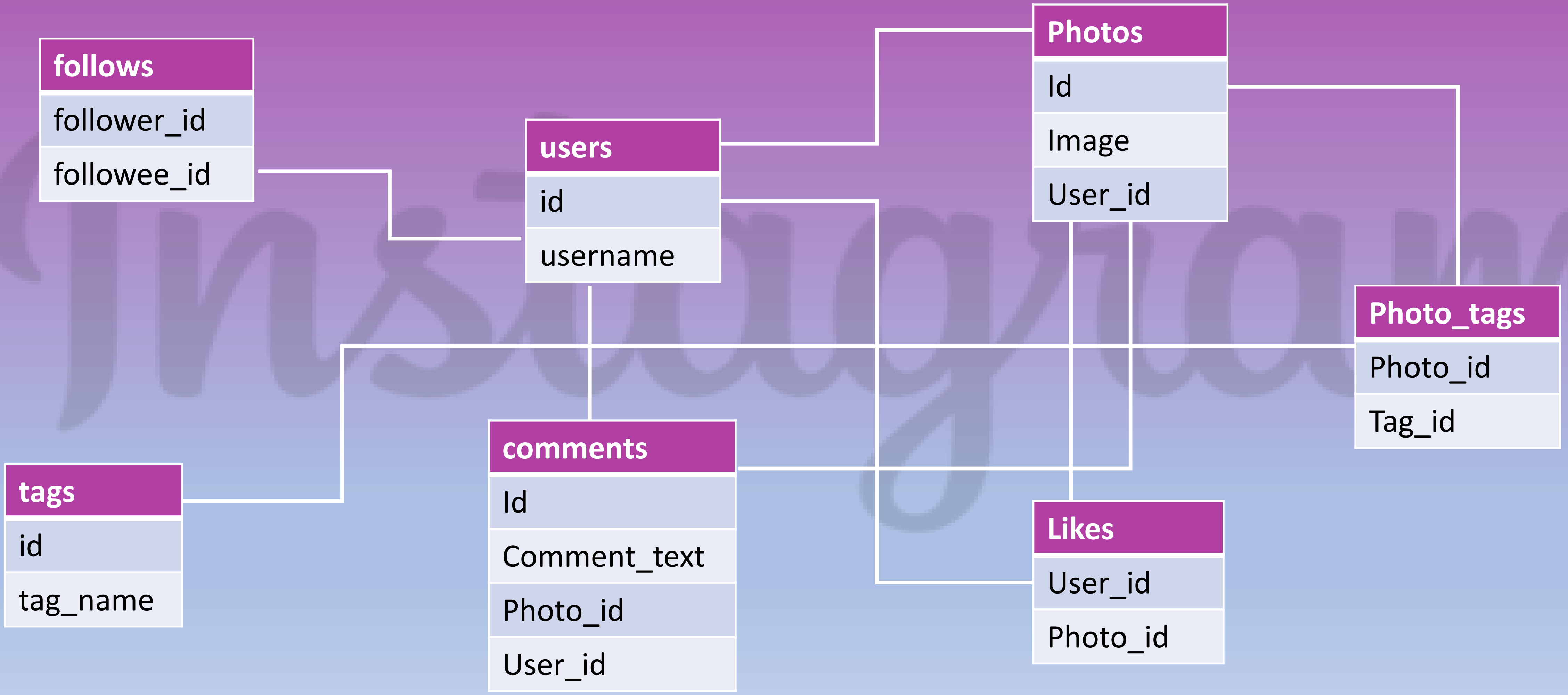
Insta Insights

- Unveiling User Dynamics and Content Strategies in the Instagram Clone Universe Using MySQL

Introduction

- As we navigate through user profiles and engagement metrics, this project aims to offer a comprehensive understanding of user interactions and unveiling strategic opportunities for the platform's advancement

Database - Instagram Clone



follows
- follower_id
- followee_id

users
- id
- username

Photos
- Id
- Image
- User_id

Photo_tags
- Photo_id
- Tag_id

tags
- id
- tag_name

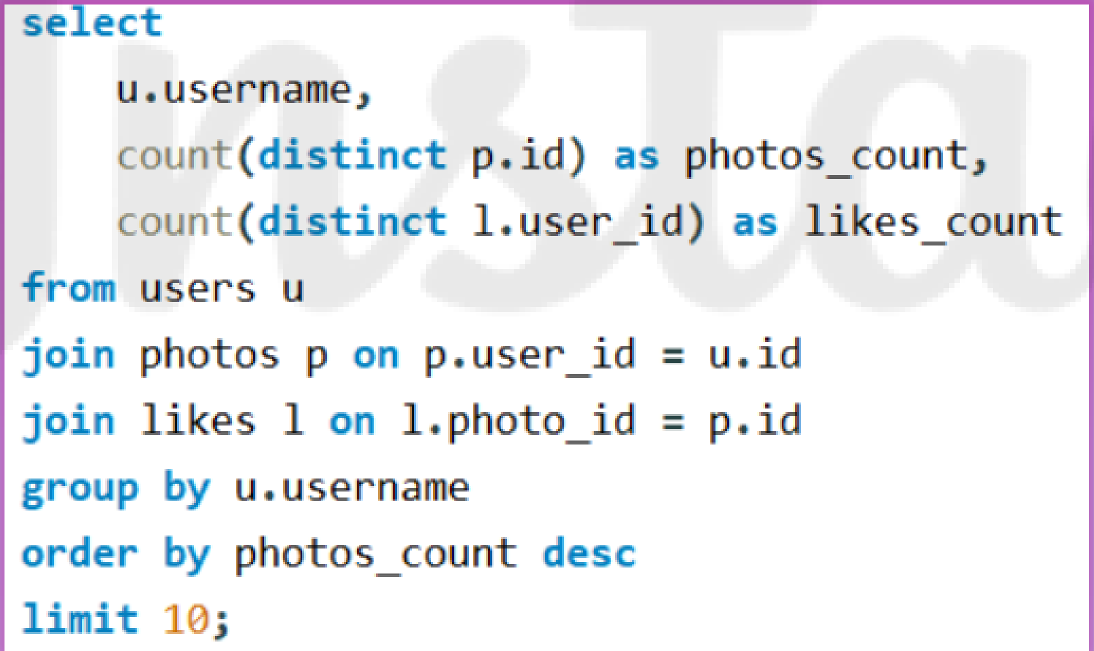
comments
- Id
- Comment_text
- Photo_id
- User_id

Likes
- User_id
- Photo_id

User Engagement Analysis

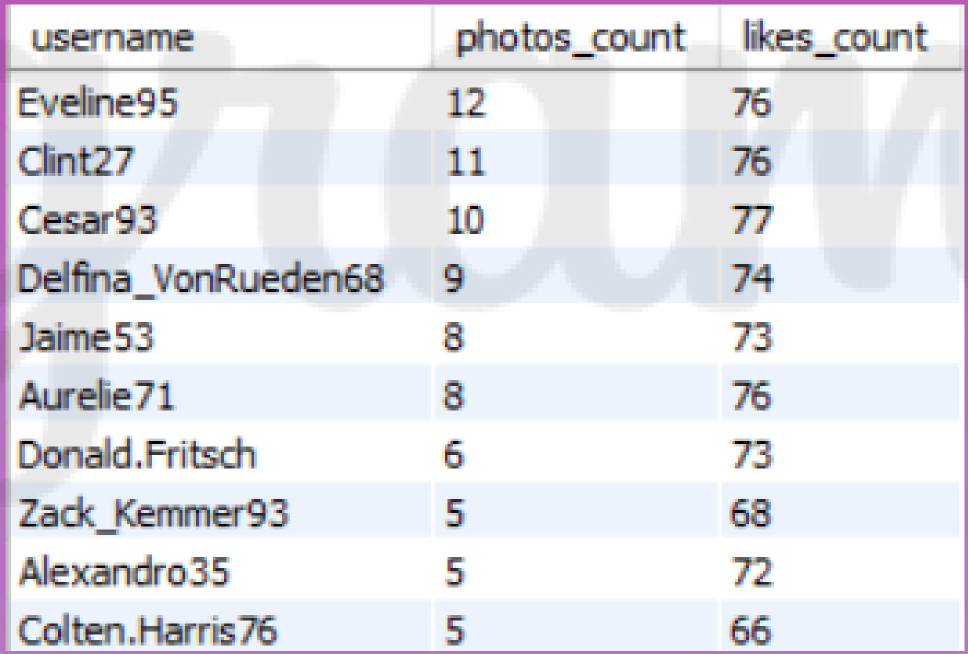
- Top 10 users with the highest number of photos and likes

MySQL Query



```sql
select
    u.username,
    count(distinct p.id) as photos_count,
    count(distinct l.user_id) as likes_count
from users u
join photos p on p.user_id = u.id
join likes l on l.photo_id = p.id
group by u.username
order by photos_count desc
limit 10;
```

Output

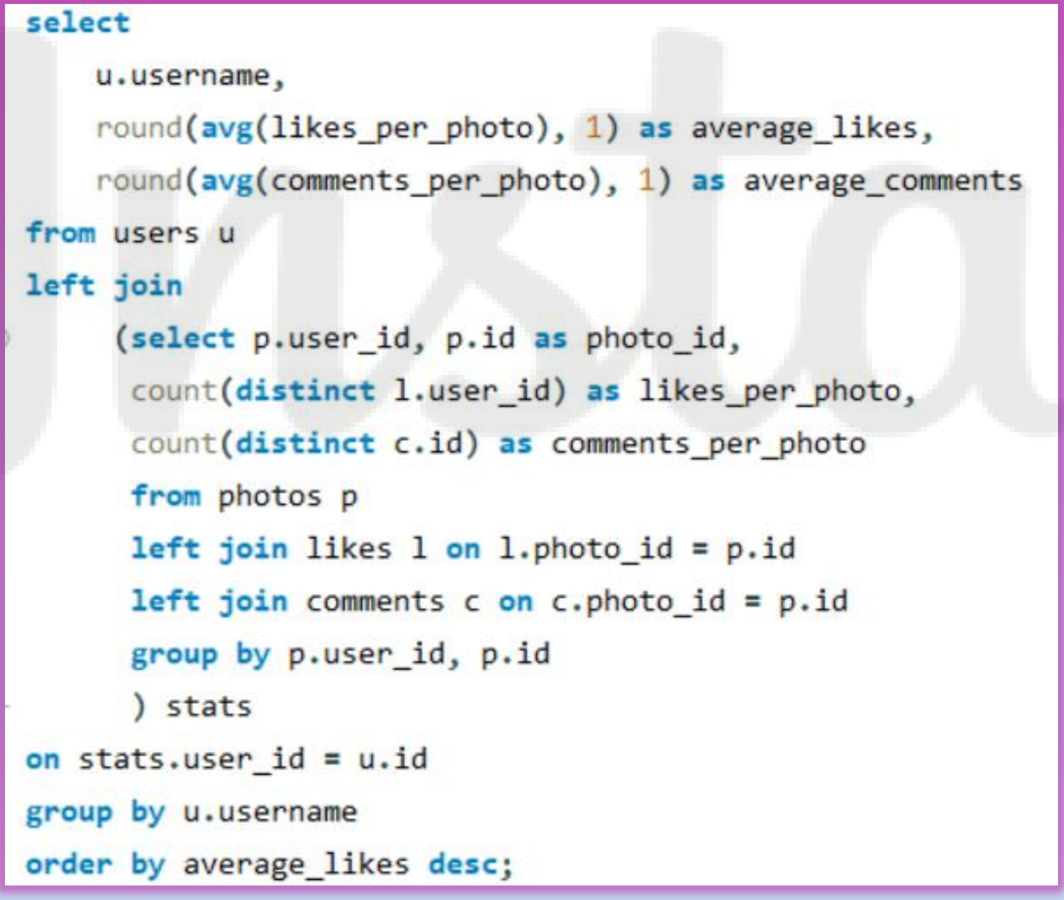


username	photos_count	likes_count
Eveline95	12	76
Clint27	11	76
Cesar93	10	77
Delfina_VonRueden68	9	74
Jaime53	8	73
Aurelie71	8	76
Donald.Fritsch	6	73
Zack_Kemmer93	5	68
Alexandro35	5	72
Colten.Harris76	5	66

User Engagement Analysis

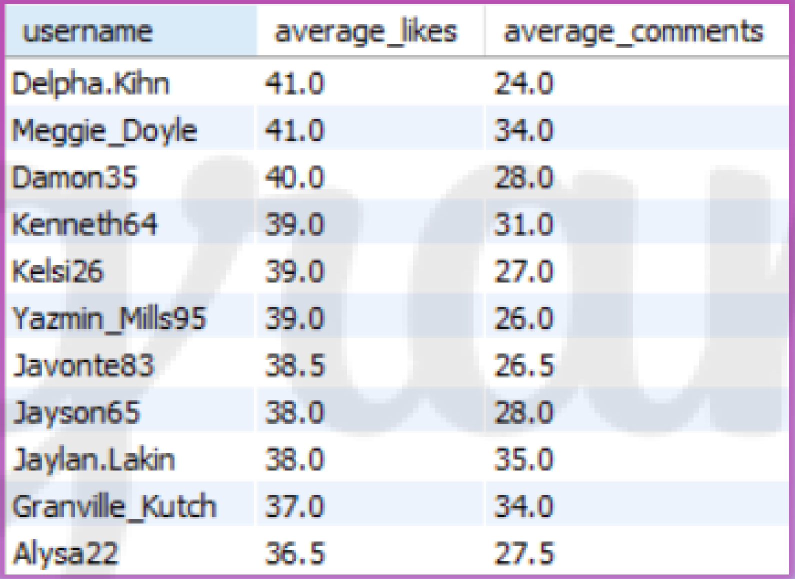
- **Average number of likes and comments per photo for each user**

MySQL Query



```
select
    u.username,
    round(avg(likes_per_photo), 1) as average_likes,
    round(avg(comments_per_photo), 1) as average_comments
from users u
left join
     (select p.user_id, p.id as photo_id,
      count(distinct l.user_id) as likes_per_photo,
      count(distinct c.id) as comments_per_photo
      from photos p
      left join likes l on l.photo_id = p.id
      left join comments c on c.photo_id = p.id
      group by p.user_id, p.id
      ) stats
on stats.user_id = u.id
group by u.username
order by average_likes desc;
```

Output

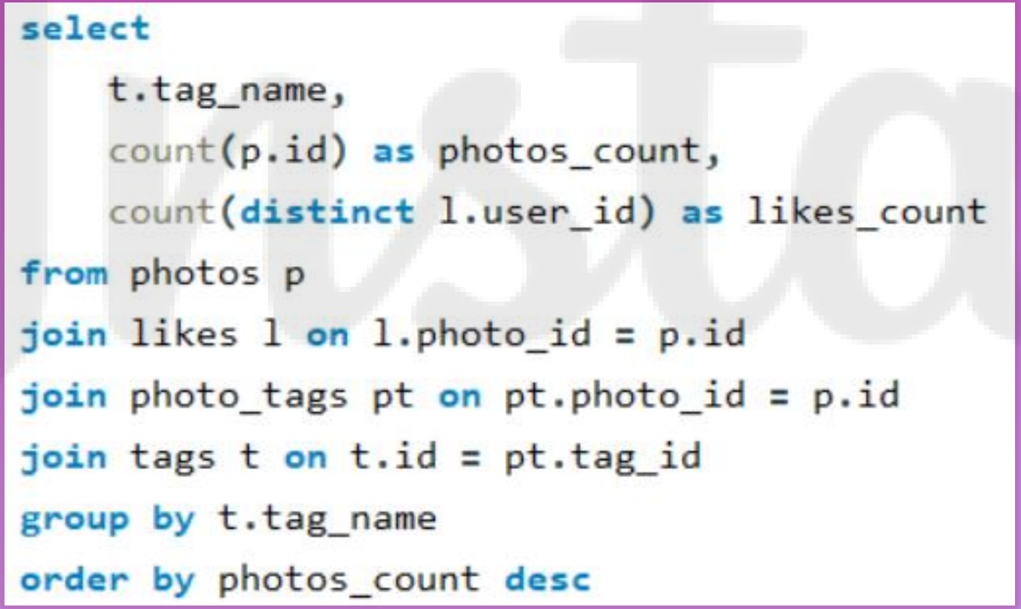


username	average_likes	average_comments
Delpha.Kihn	41.0	24.0
Meggie_Doyle	41.0	34.0
Damon35	40.0	28.0
Kenneth64	39.0	31.0
Kelsi26	39.0	27.0
Yazmin_Mills95	39.0	26.0
Javonte83	38.5	26.5
Jayson65	38.0	28.0
Jaylan.Lakin	38.0	35.0
Granville_Kutch	37.0	34.0
Alysa22	36.5	27.5

User Engagement Analysis

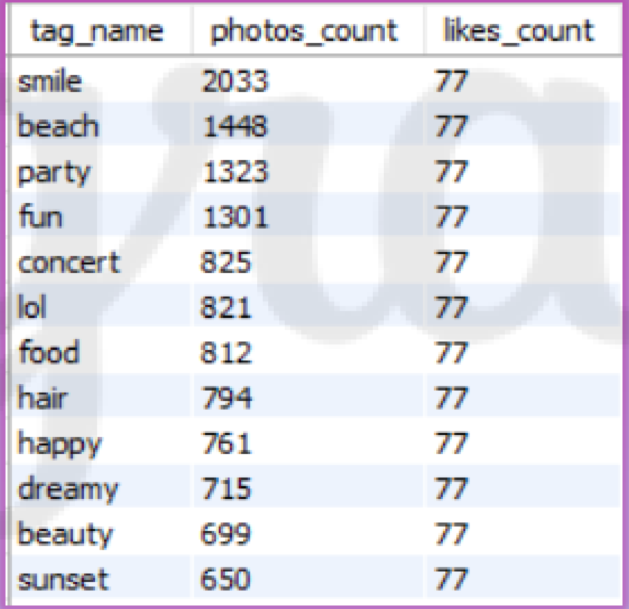
- Most popular tags based on the number of photos and likes

MySQL Query



```
select
    t.tag_name,
    count(p.id) as photos_count,
    count(distinct l.user_id) as likes_count
from photos p
join likes l on l.photo_id = p.id
join photo_tags pt on pt.photo_id = p.id
join tags t on t.id = pt.tag_id
group by t.tag_name
order by photos_count desc
```

Output

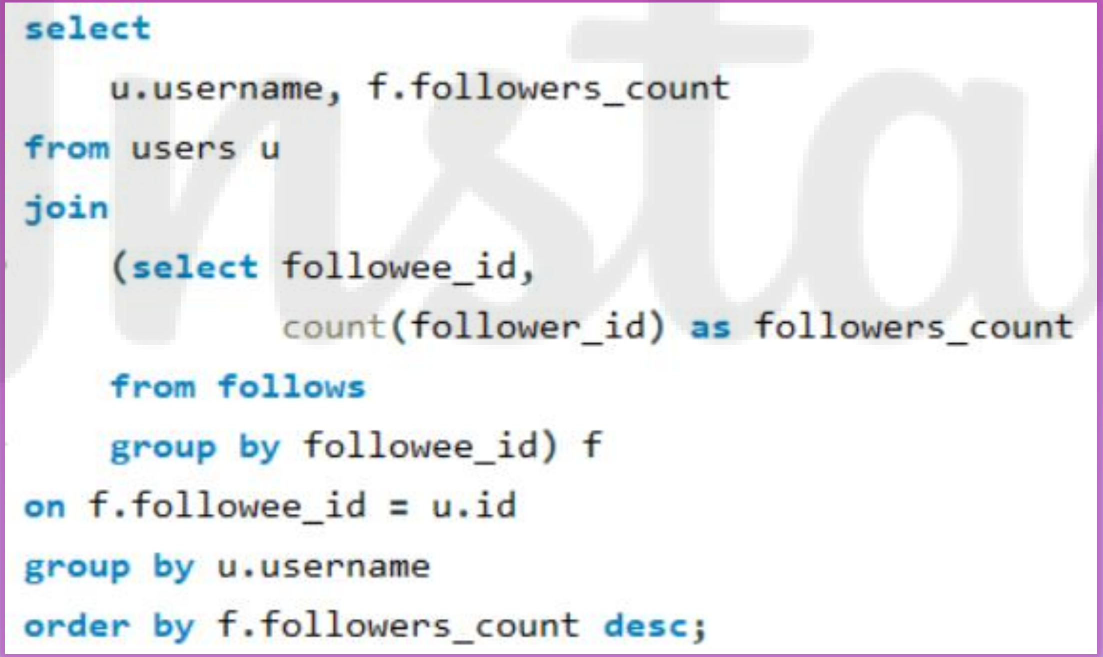


tag_name	photos_count	likes_count
smile	2033	77
beach	1448	77
party	1323	77
fun	1301	77
concert	825	77
lol	821	77
food	812	77
hair	794	77
happy	761	77
dreamy	715	77
beauty	699	77
sunset	650	77

User Interaction and Relationships

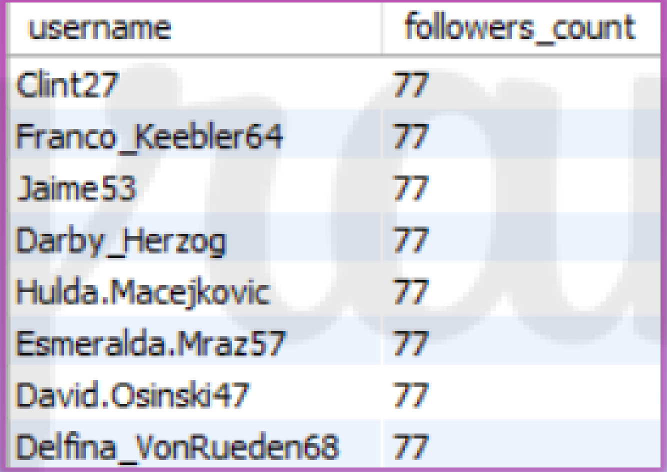
- Users who have the highest number of followers

MySQL Query



```
select
    u.username, f.followers_count
from users u
join
    (select followee_id,
            count(follower_id) as followers_count
    from follows
    group by followee_id) f
on f.followee_id = u.id
group by u.username
order by f.followers_count desc;
```

Output

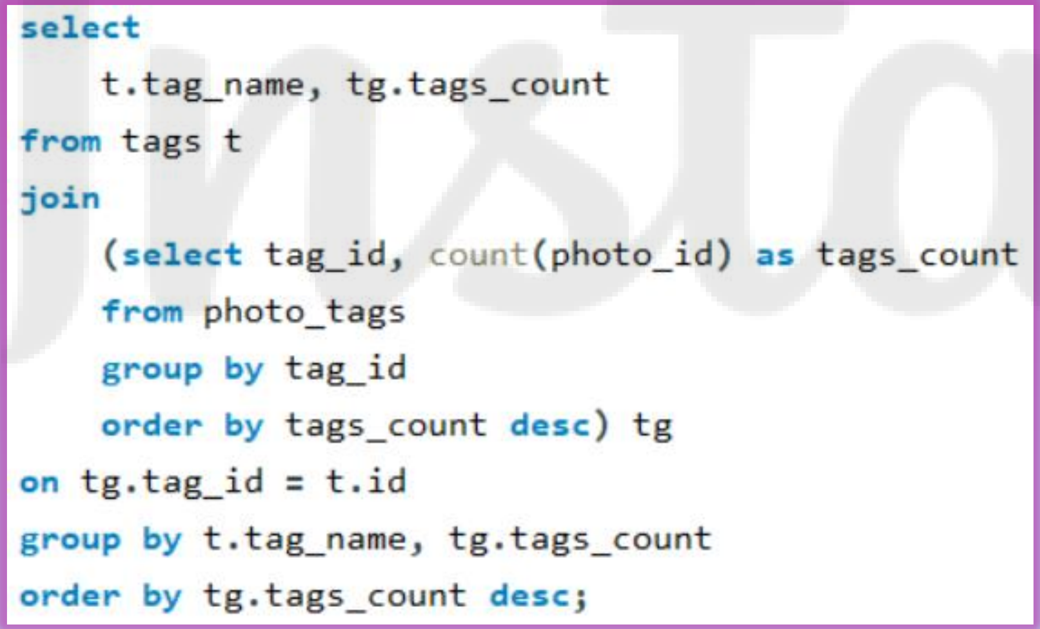


username	followers_count
Clint27	77
Franco_Keebler64	77
Jaime53	77
Darby_Herzog	77
Hulda.Macejkovic	77
Esmeralda.Mraz57	77
David.Osinski47	77
Delfina_VonRueden68	77

Tag Analysis

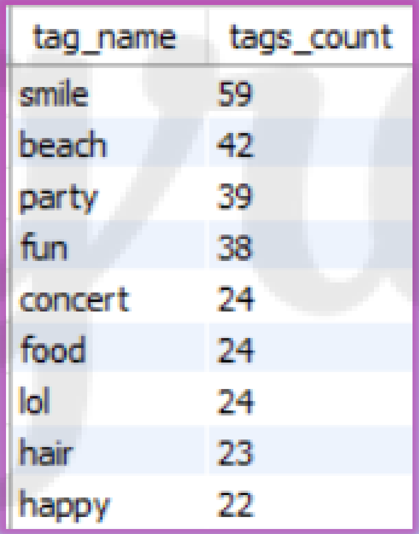
- Most commonly used tags

MySQL Query



```
select
    t.tag_name, tg.tags_count
from tags t
join
    (select tag_id, count(photo_id) as tags_count
    from photo_tags
    group by tag_id
    order by tags_count desc) tg
on tg.tag_id = t.id
group by t.tag_name, tg.tags_count
order by tg.tags_count desc;
```

Output

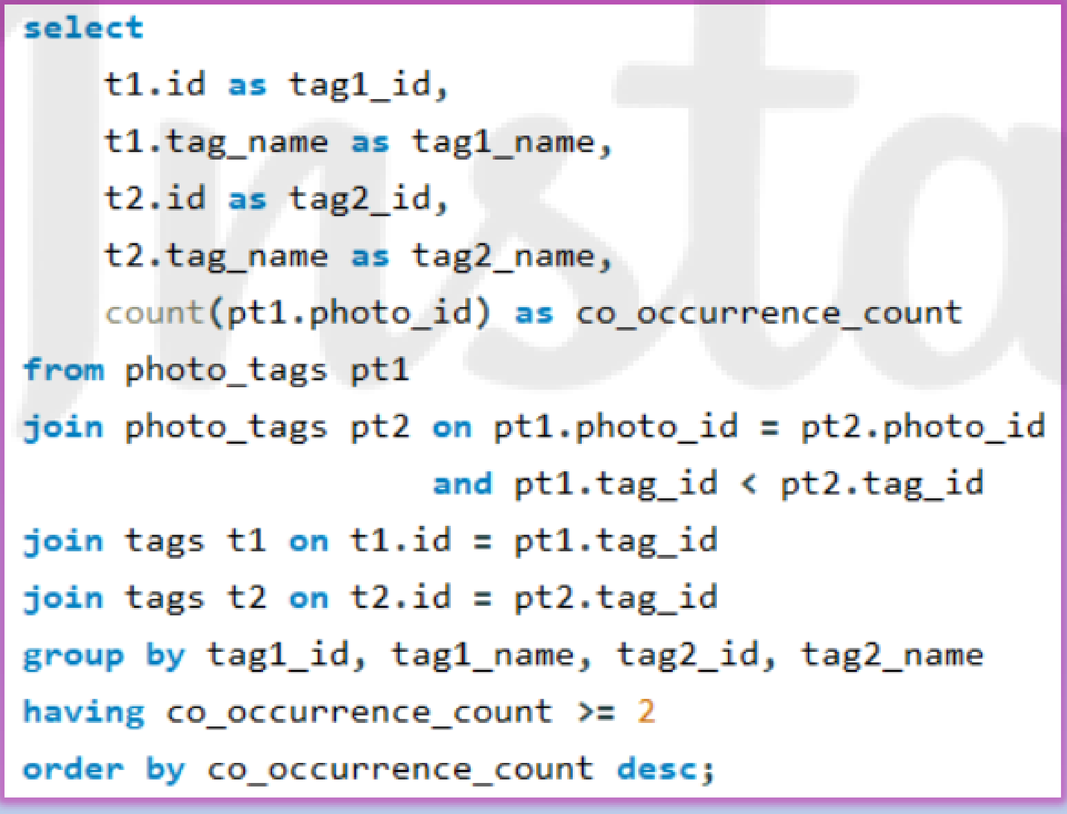


tag_name	tags_count
smile	59
beach	42
party	39
fun	38
concert	24
food	24
lol	24
hair	23
happy	22

Tag Analysis

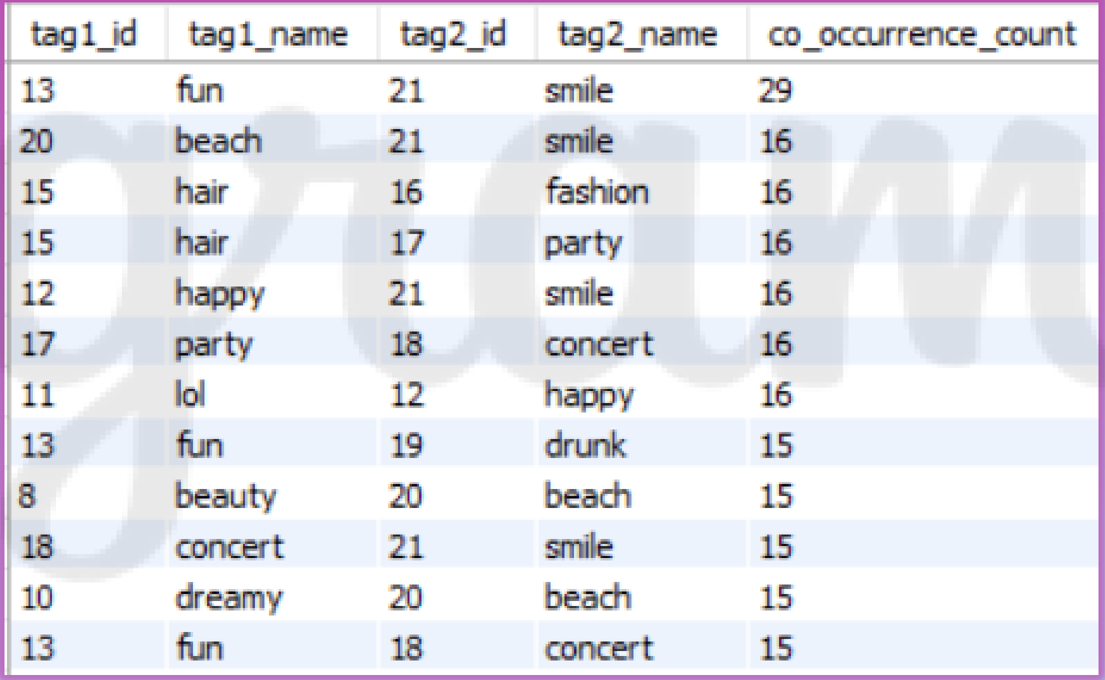
- tags that are frequently used together
- Per analysis, #Fun and #Smile are suggested tag bundles

MySQL Query



```
select
    t1.id as tag1_id,
    t1.tag_name as tag1_name,
    t2.id as tag2_id,
    t2.tag_name as tag2_name,
    count(pt1.photo_id) as co_occurrence_count
from photo_tags pt1
join photo_tags pt2 on pt1.photo_id = pt2.photo_id
                    and pt1.tag_id < pt2.tag_id
join tags t1 on t1.id = pt1.tag_id
join tags t2 on t2.id = pt2.tag_id
group by tag1_id, tag1_name, tag2_id, tag2_name
having co_occurrence_count >= 2
order by co_occurrence_count desc;
```

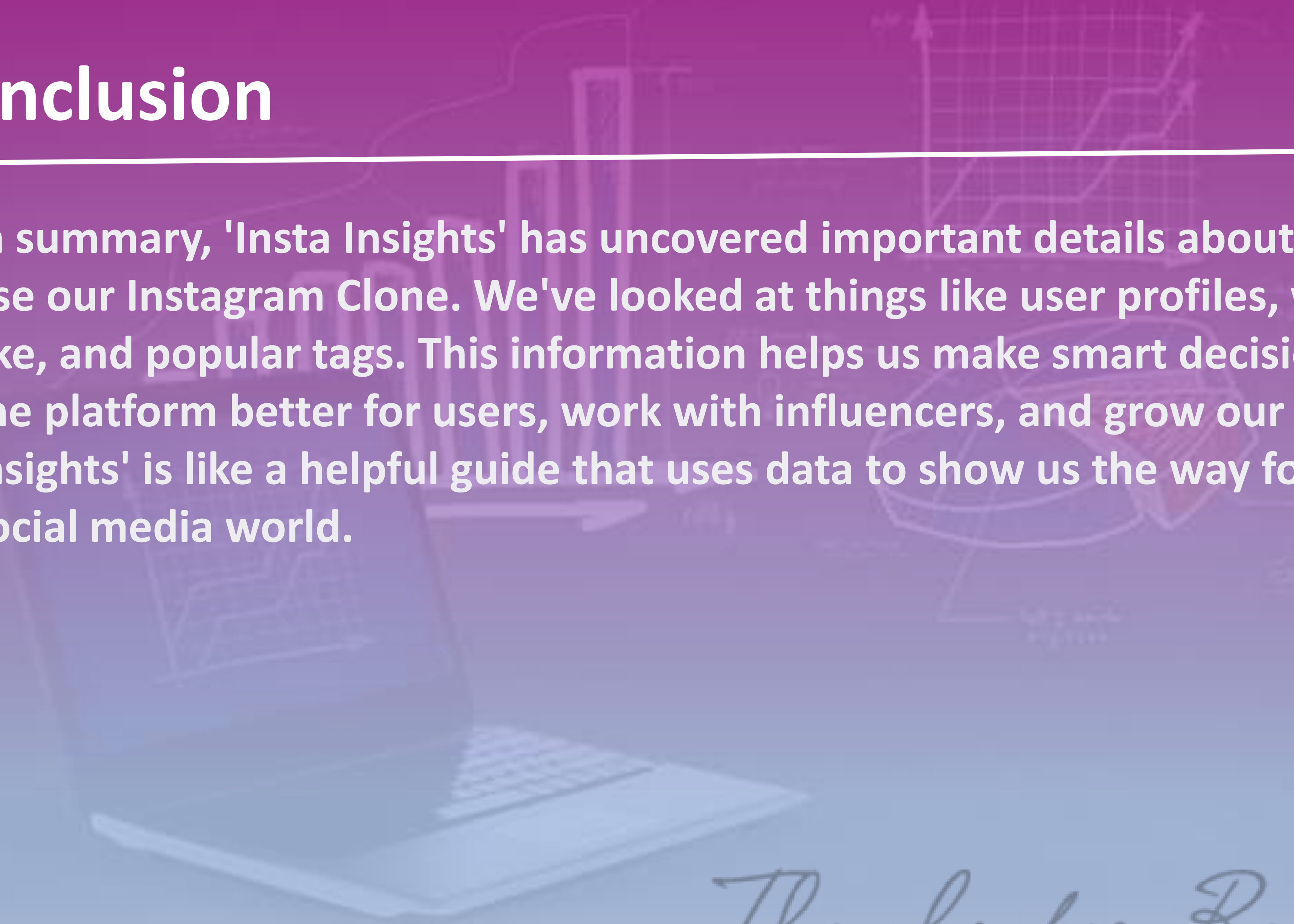
Output



tag1_id	tag1_name	tag2_id	tag2_name	co_occurrence_count
13	fun	21	smile	29
20	beach	21	smile	16
15	hair	16	fashion	16
15	hair	17	party	16
12	happy	21	smile	16
17	party	18	concert	16
11	lol	12	happy	16
13	fun	19	drunk	15
8	beauty	20	beach	15
18	concert	21	smile	15
10	dreamy	20	beach	15
13	fun	18	concert	15

Conclusion

- In summary, 'Insta Insights' has uncovered important details about how people use our Instagram Clone. We've looked at things like user profiles, what people like, and popular tags. This information helps us make smart decisions to make the platform better for users, work with influencers, and grow our business. 'Insta Insights' is like a helpful guide that uses data to show us the way forward for our social media world.



Thanks for Reading!