Instagram User Analytics

Project Description:

In this project, I have to to use SQL and MySQL Workbench to study how users interact with the app. This involves tracking user behaviour to provide insights. These insights are valuable for teams like marketing, who can use them for campaigns, product teams to decide on new features, and development teams to enhance user experience. The goal is to make informed decisions about Instagram's future development, potentially influencing one of the world's most popular social media platforms.

Approach:

To effectively analyze Instagram user data using SQL and MySQL Workbench, my approach would start with understanding the specific questions posed by the management team. I'll identify key metrics such as user engagement, feature popularity, and behavioral patterns. Leveraging SQL queries, I'll extract relevant data, conduct exploratory analysis, and generate meaningful insights.

Tech-Stack Used:

MySQL 8.0 Workbench

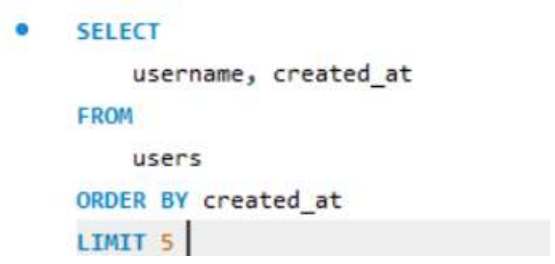
Insights

By effectively analyzing the data I have got few insights

A) Marketing

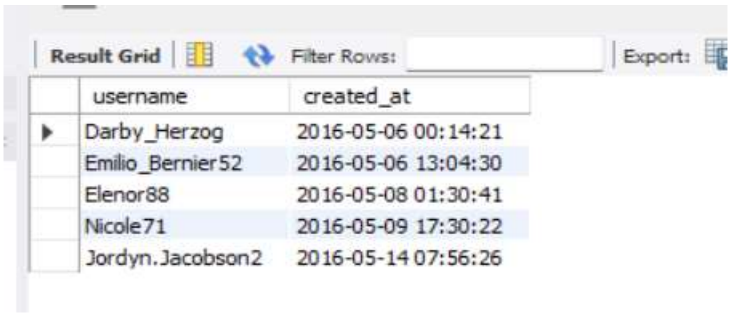
Task 1:

In order to reward most loyal users, find the 5 oldest users of the Instagram from the database provided.



```
SELECT
    username, created_at
FROM
    users
ORDER BY created_at
LIMIT 5
```

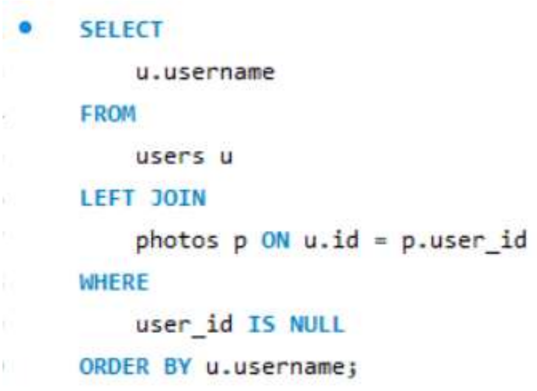
Output:



username	created_at
Darby_Herzog	2016-05-06 00:14:21
Emilio_Bernier52	2016-05-06 13:04:30
Elenor88	2016-05-08 01:30:41
Nicole71	2016-05-09 17:30:22
Jordyn.Jacobson2	2016-05-14 07:56:26

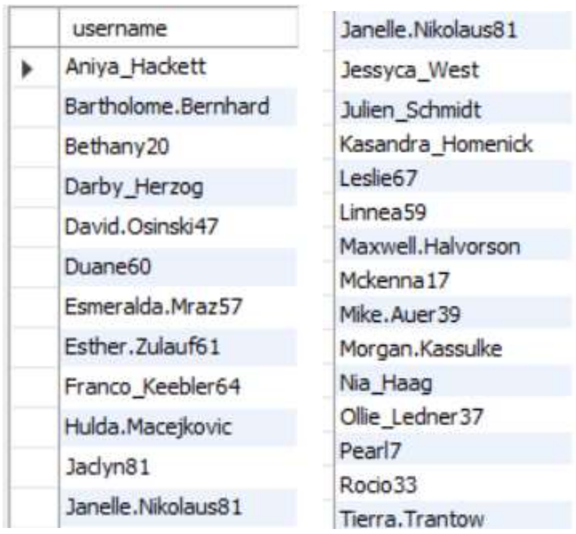
Task 2:

Find the users who have never posted a single photo on Instagram.



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

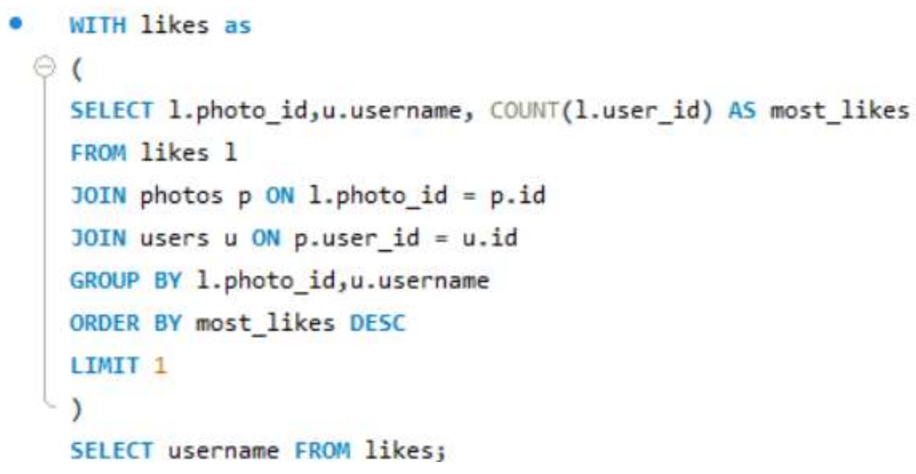
Output:



username
Aniya_Hackett
Bartholome.Bernhard
Bethany20
Darby_Herzog
David.Osinski47
Duane60
Esmeralda.Mraz57
Esther.Zulauf61
Franco_Keebler64
Hulda.Macejkovic
Jaclyn81
Janelle.Nikolaus81
Janelle.Nikolaus81
Jessyca_West
Julien_Schmidt
Kasandra_Homenick
Leslie67
Linnea59
Maxwell.Halvorson
Mckenna17
Mike.Auer39
Morgan.Kassulke
Nia_Haag
Ollie_Ledner37
Pearl7
Rocio33
Tierra.Trantow

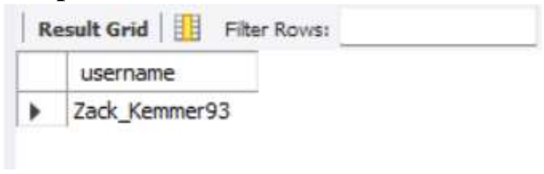
Task 3:

Identify the winner of the contest by finding out which user gets the most likes on a single photo and provide their details to the team.



```
WITH likes as
(
SELECT l.photo_id,u.username, COUNT(l.user_id) AS most_likes
FROM likes l
JOIN photos p ON l.photo_id = p.id
JOIN users u ON p.user_id = u.id
GROUP BY l.photo_id,u.username
ORDER BY most_likes DESC
LIMIT 1
)
SELECT username FROM likes;
```

Output:



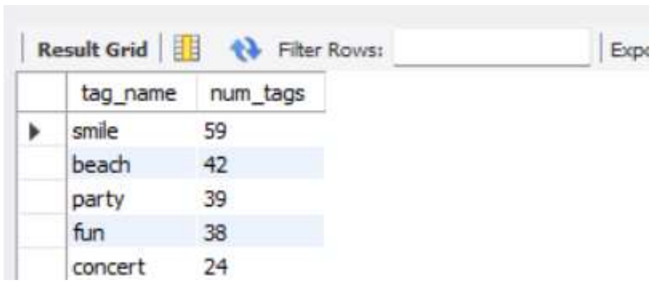
	username
▶	Zack_Kemmer93

Task 4:

Identify and suggest the top 5 most commonly used hashtags on the platform.

```
SELECT
    t.tag_name, COUNT(p.photo_id) AS num_tags
FROM
    tags t
JOIN
    photo_tags p ON t.id = p.tag_id
GROUP BY t.tag_name
ORDER BY num_tags DESC
LIMIT 5;
```

Output:



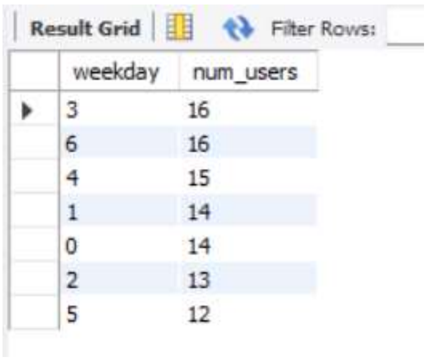
	tag_name	num_tags
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

Task 5:

What day of the week do most users register on? Provide insights on when to schedule an ad campaign.

```
SELECT
    WEEKDAY(created_at) AS weekday, COUNT(username) AS num_users
FROM
    users
GROUP BY 1
ORDER BY 2 DESC
```

Output:



weekday	num_users
3	16
6	16
4	15
1	14
0	14
2	13
5	12

If we find the date by using created_at, we will come to know the day corresponding to the dates. Here, 0 – Monday, 1 – Tuesday, 2 – Wednesday, 3 – Thursday, 4 – Friday, 5 –Saturday, 6 – Sunday

We can see maximum number of users are registering on 3 and 6 i.e. Thursday and Sunday

B) Investors metric

Task 6:

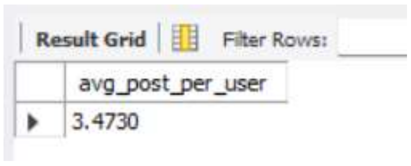
Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users.

- Average user post on instagram

```
-- Average_post_per_user
WITH post as
(
SELECT u.id AS user_id,
COUNT(p.id) AS photoid
FROM users u
LEFT JOIN photos p
ON u.id = p.user_id
GROUP BY u.id
)

SELECT SUM(photoid)/count(user_id) AS avg_post_per_user
FROM post
WHERE photoid > 0;
```

Output:



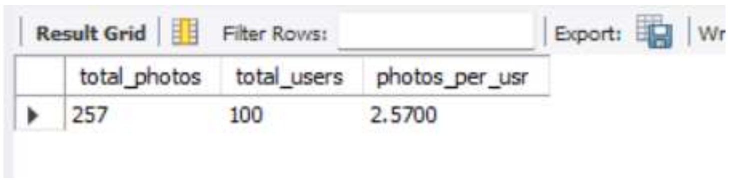
```
avg_post_per_user
3.4730
```

- Average Photos per user

```
-- Average photo per user
WITH post as
(
SELECT u.id AS user_id,
COUNT(p.id) AS photoid
FROM users u
LEFT JOIN photos p
ON u.id = p.user_id
GROUP BY u.id
)

SELECT SUM(photoid) AS total_photos,
count(user_id) AS total_users,
SUM(photoid)/count(user_id) AS photos_per_usr
FROM post;
```

Output:


Result Grid | Filter Rows: | Export: | Wr

	total_photos	total_users	photos_per_usr
▶	257	100	2.5700

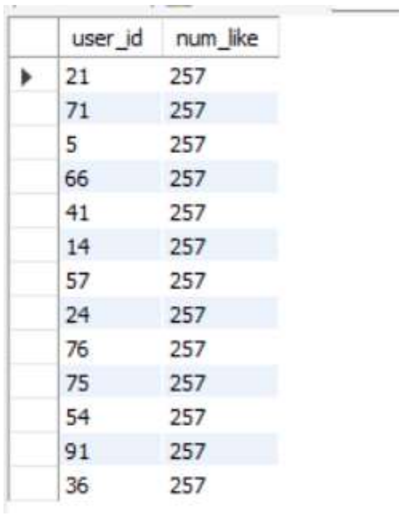
Task 7:

Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).

```
WITH photo_count AS
(
SELECT user_id,
COUNT(photo_id) AS num_like
FROM likes
GROUP BY user_id
ORDER BY num_like DESC
)

SELECT *FROM photo_count
WHERE
num_like = ( SELECT count(*) FROM photos)
```

Output:



	user_id	num_like
▶	21	257
	71	257
	5	257
	66	257
	41	257
	14	257
	57	257
	24	257
	76	257
	75	257
	54	257
	91	257
	36	257

As any normal person won’t like every single photo on instagram, I conclude that these users are bots.

Conclusion:

In conclusion, these SQL analyses contribute valuable insights for the marketing team to recognize loyal users, encourage inactive users, declare contest winners, and optimize content strategy. Simultaneously, investor metrics focus on user engagement and the identification of potential fake accounts, offering a comprehensive view of Instagram's health and authenticity for potential investors.