

Instagram User Analytics

Project Description: This project aims to extract useful insights from raw data or meta data using various database management tools and even visualize them to increase the platform efficiency.

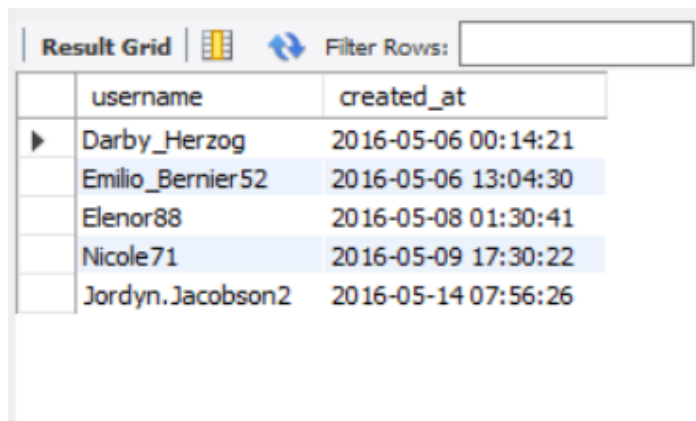
A) Marketing Analysis:

1. **Loyal User Reward:** Those who have been using the platform for the longest time.

Code: select * from users;

select username,created_at from users order by created_at LIMIT 5;

Output:



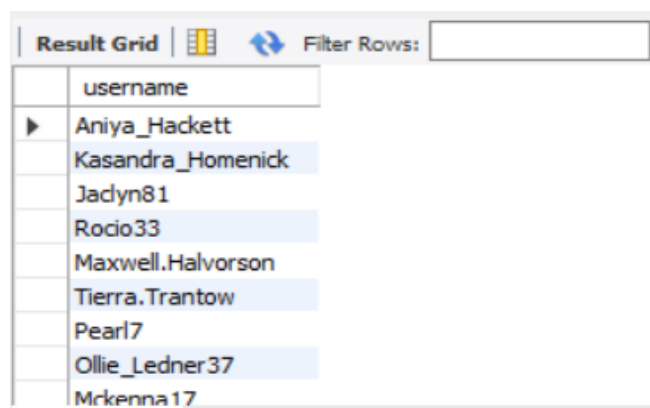
The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a query that selects the top 5 users by their creation date. The table has two columns: 'username' and 'created_at'. The data is as follows:

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

2. **Inactive User Engagement:** Identify users who have never posted a single photo on Instagram.

Code: select * from photos;

select username from users left join photos on users.id=photos.user_id
where photos.id is null;



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a query that identifies users who have never posted a photo. The table has one column: 'username'. The data is as follows:

username
Aniya_Hackett
Kassandra_Homenick
Jadyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mrkenna17

Result Grid	Filter Rows:
username	
Ollie_Ledner37	
Mckenna17	
David.Osinski47	
Morgan.Kassulke	
Linnea59	
Duane60	
Julien_Schmidt	
Mike.Auer39	
Franco_Keebler64	

Result Grid	Filter Rows:
username	
Franco_Keebler64	
Nia_Haag	
Hulda.Macejkovic	
Leslie67	
Janelle.Nikolaus81	
Darby_Herzog	
Esther.Zulauf61	
Bartholome.Bernhard	
Jessyca_West	

Result 20 x

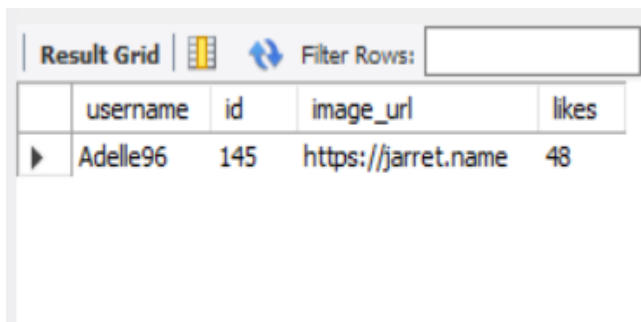
Result Grid	Filter Rows:
username	
Leslie67	
Janelle.Nikolaus81	
Darby_Herzog	
Esther.Zulauf61	
Bartholome.Bernhard	
Jessyca_West	
Esmeralda.Mraz57	
Bethany20	

3. Contest Winner Declaration: Determine the winner of the contest and provide their details to the team.

Code: select * from likes, photos, users;

select username, photos.id, photos.image_url, count(likes.user_id) as
likes from likes inner join photos on likes.photo_id = photos.id inner join
users on photos.user_id group by likes.photo_id, users.username order by
likes desc limit 1;

Output:



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' input field and a table with the following data:

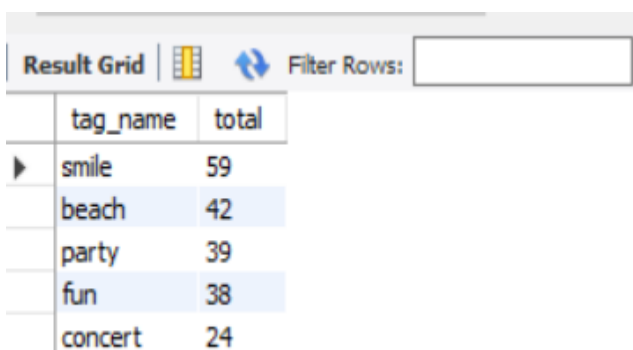
	username	id	image_url	likes
▶	Adelle96	145	https://jarret.name	48

- 4. Hashtag Research:** Identify and suggest the top five most used hashtags on the platform.

Code: select * from tags;

select tags.tag_name, count(*) as total from photo_tags join tags on photo_tags.tag_id=tags.id group by tags.id order by total desc limit 5;

Output:



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' input field and a table with the following data:

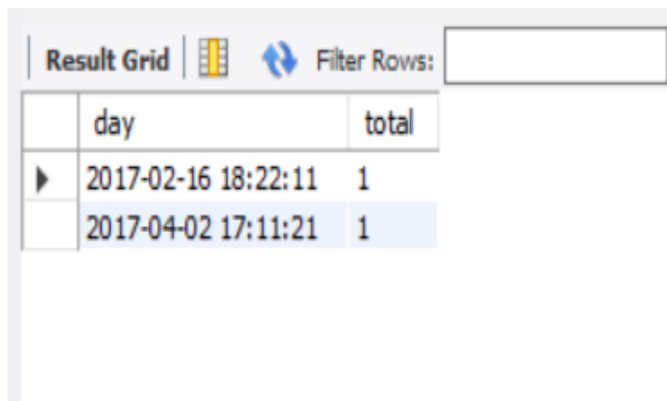
	tag_name	total
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

- 5. Ad Campaign Launch:** Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

Code: select* from users;

select(created_at) as day,count(*) as total from users group by day order by total desc limit 2;

Output:



The screenshot shows a database interface with a 'Result Grid' tab. The grid has two columns: 'day' and 'total'. The first row shows the date '2017-02-16 18:22:11' with a total of '1'. The second row shows the date '2017-04-02 17:11:21' with a total of '1'. There is a 'Filter Rows' input field to the right of the grid.

	day	total
▶	2017-02-16 18:22:11	1
	2017-04-02 17:11:21	1

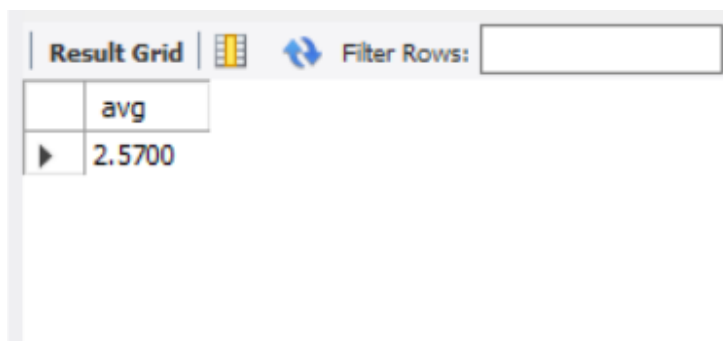
B) Investor Metrics:

- 1. User Engagement:** Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

Code: select * from photos, users;

select (select count (*) from photos)/ (select count (*) from users) as avg;

Output:



The screenshot shows a database interface with a 'Result Grid' tab. The grid has one column: 'avg'. The first row shows the value '2.5700'. There is a 'Filter Rows' input field to the right of the grid.

	avg
▶	2.5700

- 2. Bots & Fake Accounts:** Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

Code: select * from users, likes;

select user_id, count(*) as num_likes from likes group by user_id having num_likes = (select count(*) from photos);

```
select u.username,count(*) as num_likes from users u join likes l on u.id = l.user_id group by u.id having num_likes = (select count(*) from photos);
```

Output:

Result Grid			Filter Rows:
	username	num_likes	
▶	Aniya_Hackett	257	
	Jadyn81	257	
	Rocio33	257	
	Maxwell.Halvorson	257	
	Ollie_Ledner37	257	
	Mckenna17	257	
	Duane60	257	
	Julien_Schmidt	257	
	Mike.Auer39	257	

Result Grid			Filter Rows:
	username	num_likes	
	Mckenna17	257	
	Duane60	257	
	Julien_Schmidt	257	
	Mike.Auer39	257	
	Nia_Haag	257	
	Leslie67	257	
	Janelle.Nikolaus81	257	
	Bethany20	257	

