

Instagram User Analytics

Submitted by – Akhilesh Chaturvedi

Project Description

The Instagram Users database was used for analytics to answer business problems as requested by the management team. The database contains seven tables with the following under mentioned details. The Marketing team, to better reward its users and design ad-campaign wants to know how users are reacting to it. Similarly, Investors also has some queries, for which an attempt has been made to address these issues.

1. Users

Field	Type	Null	Key	Default	Extra
Id	int	NO	PRI		auto_increment
Username	varchar(255)	NO			
created_at	timestamp	YES		CURRENT_TIME STAMP	DEFAULT_GENE RATED

2. likes

Field	Type	Null	Key	Default	Extra
user_id	Int	NO	PRI		
photo_id	Int	NO	PRI		
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

3. photos

Field	Type	Null	Key	Default	Extra
Id	Int	NO	PRI		auto_increment
image_url	varchar(355)	NO			
user_id	Int	NO	MUL		
created_dat	timestamp	YES		CURRENT_TIM ESTAMP	DEFAULT_GENE RATED

4. photo_tags

Field	Type	Null	Key	Default	Extra
photo_id	int	NO	PRI		
tag_id	int	NO	PRI		

5. tags

Field	Type	Null	Key	Default	Extra
Id	Int	NO	PRI		auto_increment
tag_name	varchar(255)	NO	UNI		
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

6. follows

Field	Type	Null	Key	Default	Extra
follower_id	int	NO	PRI		
followee_id	int	NO	PRI		
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

7. Comments

Field	Type	Null	Key	Default	Extra
'id'	'int'	'NO'	'PRI'	NULL	'auto_increment'
'comment_text'	'varchar(255)'	'NO'	'"	NULL	'"
'user_id'	'int'	'NO'	'MUL'	NULL	'"
'photo_id'	'int'	'NO'	'MUL'	NULL	'"
'created_at'	'timestamp'	'YES'	'"	'CURRENT_TIMESTAMP'	'DEFAULT_GENERATED'

Approach

The database named “ig_clone” was created in MySQL using provided code snippet. Then Analytics was performed according to the respective Problem statements and SQL queries, associated with the problem are mentioned below in the report. Understanding of the tables is of outmost importance, as it will provide insight into information, we will require to answer the business problems.

Tech-Stack Used: MySQL Workbench 8

Problem Statements

You are required to provide a detailed report answering the questions below:

A) Marketing: The marketing team wants to launch some campaigns, and they need your help with the following

- 1. Rewarding Most Loyal Users:** People who have been using the platform for the longest time.
Your Task: Find the 5 oldest users of Instagram from the database provided

Solution:

```
SELECT username, created_at
FROM users
order by created_at asc
limit 5;
```

The “users” table contains the date of creation of the account by the users and the username. When we arrange the table in ascending order, we will get the users who are the oldest on Instagram. Following is the username of the 5 oldest users.

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

- 2. Remind Inactive Users to Start Posting:** By sending them promotional emails to post their 1st photo.
Your Task: Find the users who have never posted a single photo on Instagram

Solution:

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

Here, we need to join the “photos” and “users” tables, particularly left join because we want all users in the table where we can put the “where” condition, to fetch users who have posted photos on Instagram.

After that, putting where condition to negate the statement, we will get the desired Instagram users who have never posted a single photo.

output

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

3. **Declaring Contest Winner:** The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner. Your Task: Identify the winner of the contest and provide their details to the team

Solution:

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

Here we have to fetch the users, who got the most number of likes on a single photo, so basically we have to deal with three tables, photos (because it contains details about the photo), likes (as it contains the number of likes) and users (as it contains the name of users).

We are using Inner join in this case, as we want matching results from all three tables and then grouping them based on particular images so that we can get result sorted on the basis of the particular photo.

Then order them in descending order to get the highest-liked photo and username.

Output

username	id	image_url	total_likes
Harley_Lind18	145	https://jarret.name	48
Andre_Purdy85	127	https://celestine.name	43
Harley_Lind18	182	https://dorcas.biz	43
Aniya_Hackett	123	http://shannon.org	42
Andre_Purdy85	30	http://kenny.com	41

4. **Hashtag Researching:** A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.

Your Task: Identify and suggest the top 5 most used hashtags on the platform

Solution:

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

To find the top 5 hashtags, we will need two tables, the “photo_tags” and “tags” tables. Then performing inner join on both tables to find distinct tags and their count. Then grouping them based on tags ‘id’. Then ordering the result in descending order to get all hashtags in sequence. Finally limiting the results to the top 5.

Output

tag_name	t_tag
Smile	59
Beach	42
Party	39
Fun	38
Concert	24

5. **Launch AD Campaign:** The team wants to know, which day would be the best day to launch ADs.
Your Task: What day of the week do most users register on? Provide insights on when to schedule an ad campaign

Solution:

```
select date_format(created_at,'%a') as day_of_week, count(*) as 't_registered'
from users
group by day_of_week
order by t_registered desc;
```

Here we need to group all the dates (on a day-of-a-week basis), on which any Instagram user has registered on the platform. For this purpose, the date function (date_format) will be used which will fetch the day and we will group all the accounts created on a day-of-a-week basis. And then ordering it to find the most suitable day to launch the ad campaign.

Output

day_of_week	t_registered
Thu	16
Sun	16
Fri	15
Tue	14
Mon	14
Wed	13
Sat	12

B) Investor Metrics: Our investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds

1. **User Engagement:** Are users still as active and post on Instagram or they are making fewer posts
Your Task: Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram and/total number of users

Solution:

```
SELECT ( SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) as
average_posts_by_user;
```

For calculating the required average we need to count all the photos and all the users. And then finally dividing them.

Output

average_posts_by_user
2.5700

2. **Bots & Fake Accounts:** The investors want to know if the platform is crowded with fake and dummy accounts

Your Task: 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).

Solution:

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

Here we will use an Inner join between the “users” table and the “likes” table. Now we will count the total number of likes in the combined table. Then group them by user_id and check with the “having” condition to figure out that the user is present in the photos table. So following are the bots or dummy accounts, who have liked every single photo.

Output

Id	Username	total_likes_by_users
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike.Auer39	257
71	Nia_Haag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

