

Instagram User Analytics:

Project Description:

The project involved analysing an Instagram database to get some meaningful insights for marketing team and investors. The primary goal was to use SQL queries to get analysis on question that could impact the overall business such as **potential fake accounts, days when most maximum users register, most used hashtags** etc.

Approach:

I started by understanding the structure and content of all the 5 tables, then I used SQL queries to get the most out of the database. I used joins, subqueries, grouping, CTEs etc to generate the desired result.

Tech-Stack Used:

I used **MySQL Workbench(8.0)** which is a graphical tool for MySQL, chosen for its user-friendly interface, efficient management of databases, and the ability to run, test, and debug queries in a **visual environment**.

Insights:

I Identified trends in user behaviour, such as **highest registration days** and the **average number of posts per user**, which can help optimize content and marketing strategies. Apart from it, I found **most popular hashtags** and **the most liked photos** that ultimately provides valuable information for **content creators** and marketing teams to increase engagement.

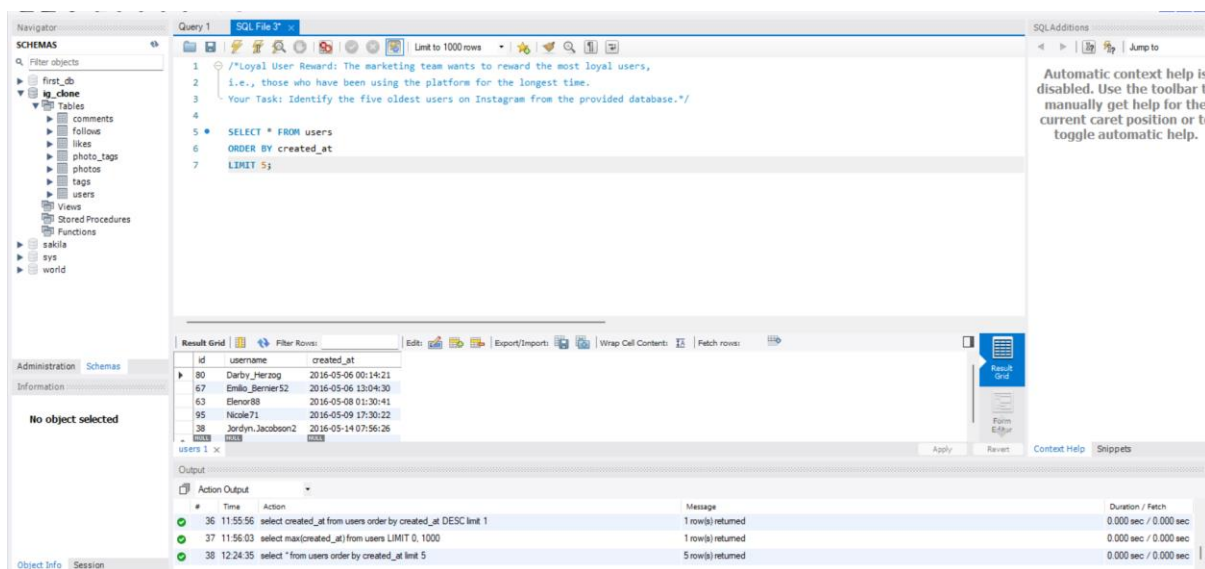
Moreover, I also got to know about **potential bots** or **fake accounts** by identifying an unusual behaviour like liking every photo on the platform. Lastly, I evaluated overall platform activity, including user engagement levels and photo uploads.

Result:

I would say this project successfully addressed all business questions, providing valuable insights for **user behaviour**, **engagement trends**, and **content popularity** on the platform. It also highlighted areas for improvement in user engagement and identified **potential issues like bot accounts**.

A) Marketing Analysis:

- **Loyal User Reward:** The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time. Your Task: Identify the five oldest users on Instagram from the provided database.



The screenshot shows a SQL IDE interface with a query editor, a result grid, and an output pane. The query editor contains the following SQL code:

```
1 /*Loyal User Reward: The marketing team wants to reward the most loyal users,  
2 i.e., those who have been using the platform for the longest time.  
3 Your Task: Identify the five oldest users on Instagram from the provided database.*/  
4  
5 SELECT * FROM users  
6 ORDER BY created_at  
7 LIMIT 5;
```

The result grid displays the following data:

id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emile_Berrier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26

The output pane shows the execution of the query, with the following messages:

```
36 11:55:56 select created_at from users order by created_at DESC limit 1 1 row(s) returned 0.000 sec / 0.000 sec  
37 11:56:03 select max(created_at) from users LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec  
38 12:24:35 select * from users order by created_at limit 5 5 row(s) returned 0.000 sec / 0.000 sec
```

- **Inactive User Engagement:** The team wants to encourage inactive users to start posting by sending them promotional emails. Your Task: Identify users who have never posted a single photo on Instagram.

Query 1

```

1  /*Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.
2  Your Task: Identify users who have never posted a single photo on Instagram.*/
3
4  * SELECT
5    users.id, users.username
6  FROM
7    users
8  LEFT JOIN
9    photos ON users.id = photos.user_id
10 WHERE
11   photos.id IS NULL;
12

```

Result Grid

id	username
5	Aniya_Hackett
7	Kassandra_Homerick
14	Jacyn81
21	Rocio33
24	Maxwell_Halvorson
42	Travis_Prasanna

Output

#	Time	Action	Message	Duration / Fetch
39	12:27:00	select users.id, users.username from users left join photos on users.id=photos.user_id where photos.id is null LI...	26 row(s) returned	0.000 sec / 0.000 sec
40	12:27:09	select users.id, users.username from users where users.id NOT IN (SELECT photos.user_id from photos) LIMIT...	26 row(s) returned	0.015 sec / 0.000 sec
41	12:27:49	SELECT users.id, users.username FROM users LEFT JOIN photos ON users.id = photos.user_id ...	26 row(s) returned	0.000 sec / 0.000 sec

- Contest Winner Declaration:** The team has organized a contest where the user with the most likes on a single photo wins.
 Your Task: Determine the winner of the contest and provide their details to the team.

Query 1

```

1  /*Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.
2  Your Task: Determine the winner of the contest and provide their details to the team.*/
3
4  * with max as (select photo_id , count(user_id) as like_count
5    from likes
6   group by photo_id
7  )
8
9  select users.username, photo_id, user_id, max.like_count, image_url from photos
10 inner join max on photos.id=max.photo_id
11 inner join users on photos.user_id=users.id
12 order by like_count DESC limit 1;
13

```

Result Grid

username	photo_id	user_id	like_count	image_url
Zack_Kenner93	145	52	48	https://jarek.name

Output

#	Time	Action	Message	Duration / Fetch
40	12:27:09	select users.id, users.username from users where users.id NOT IN (SELECT photos.user_id from photos) LIMIT...	26 row(s) returned	0.015 sec / 0.000 sec
41	12:27:49	SELECT users.id, users.username FROM users LEFT JOIN photos ON users.id = photos.user_id ...	26 row(s) returned	0.000 sec / 0.000 sec
42	12:28:25	with max as (select photo_id , count(user_id) as like_count from likes group by photo_id) select users.usa...	1 row(s) returned	0.000 sec / 0.000 sec

- Hashtag Research:** A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
 Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query designed to find the top five most common hashtags. The query uses a Common Table Expression (CTE) to first calculate the count of each tag, then joins this back to the 'tags' table to retrieve the tag names, ordered by count in descending order and limited to five results.

```

1  /*Hashtag Research: A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
2  Your Task: Identify and suggest the top five most commonly used hashtags on the platform. ( 3 tags have same value )*/
3
4  with max_tag_count as (select count(tag_id) as tag_count, tag_id
5                        from photo_tags
6                        group by tag_id
7                        )
8
9  select id, tag_name, tag_count
10 from tags
11 inner join max_tag_count on tags.id = max_tag_count.tag_id
12 order by tag_count DESC limit 5;
13

```

The 'Result Grid' shows the output of the query, displaying five rows of data with columns 'id', 'tag_name', and 'tag_count'.

id	tag_name	tag_count
21	smile	59
20	beach	42
17	party	39
13	fun	38
5	food	24

The 'Output' tab at the bottom shows the execution log, indicating that the query was successfully executed and returned 5 rows.

- Ad Campaign Launch:** The team wants to know the best day of the week to launch ads.
 Your Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

File Edit View Query Database Server Tools Scripting Help

Navigator

HEMAS

Filter objects

first_db

ig_clone

Tables

comments

follows

likes

photo_tags

photos

tags

users

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Query 1 SQL File 3'

Limit to 1000 rows

1 /*Ad Campaign Launch: The team wants to know the best day of the week to launch ads.

2 Your Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.*//

3

4

5 SELECT

6 DAYNAME(created_at) AS Day, COUNT(*) AS count_of_users

7 FROM

8 users

9 GROUP BY DAYNAME(created_at)

10 ORDER BY count_of_users DESC;

Result Grid

Filter Rows

Exports

Wrap Cell Contents

Day count_of_users

Thursday 16

Sunday 15

Friday 15

Tuesday 14

Monday 14

Wednesday 13

Result 8

Read Only

Context Help

Snippets

Output

Action Output

Time Action Message Duration / Fetch

45 12:31:11 SELECT DAYNAME(created_at) AS Day, COUNT(*) AS count_of_users FROM users GROUP BY DAYNAME(created_at) 2 row(s) returned 0.000 sec / 0.000 sec

46 12:31:18 SELECT DAYNAME(created_at) AS Day, COUNT(*) AS count_of_users FROM users GROUP BY DAYNAME(created_at) 3 row(s) returned 0.000 sec / 0.000 sec

47 12:31:43 SELECT DAYNAME(created_at) AS Day, COUNT(*) AS count_of_users FROM users GROUP BY DAYNAME(created_at) 7 row(s) returned 0.000 sec / 0.000 sec

B) Investor Metrics:

- User Engagement:** Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
 Your Task: 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.

File Edit View Query Database Server Tools Scripting Help

Navigator

HEMAS

Filter objects

first_db

ig_clone

Tables

comments

follows

likes

photo_tags

photos

tags

users

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Query 1 SQL File 3'

Limit to 1000 rows

1 /*User Engagement: Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

2 Your Task: Calculate the average number of posts per user on Instagram.

3 Also, provide the total number of photos on Instagram divided by the total number of users.*//

4

5 select (select count(*) as total_photos_published from photos) / (select count(*) as total_users from users) as avg_posts_per_user;

Result Grid

Filter Rows

Exports

Wrap Cell Contents

avg_posts_per_user

2.5700

Result 10

Read Only

Context Help

Snippets

Output

Action Output

Time Action Message Duration / Fetch

47 12:31:43 SELECT DAYNAME(created_at) AS Day, COUNT(*) AS count_of_users FROM users GROUP BY DAYNAME(created_at) 7 row(s) returned 0.000 sec / 0.000 sec

48 12:33:10 select (select count(*) as total_photos_published from photos) / (select count(*) as total_users from users) as avg_posts_per_user; 1 row(s) returned 0.016 sec / 0.000 sec

49 12:34:07 select (select count(*) as total_photos_published from photos) / (select count(*) as total_users from users) as avg_posts_per_user; 1 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

PALASHSECU 1.74%

Search

ENG IN

12:34

29-08-2024

- **Bots & Fake Accounts:** Investors want to know if the platform is crowded with fake and dummy accounts.
Your Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

The screenshot shows a SQL IDE interface with a query editor, a result grid, and an action output pane.

Query 1:

```
1 /*Bots & Fake Accounts: Investors want to know if the platform is crowded with fake and dummy accounts.*/
2 Your Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.*/
3
4 with pp as (select count(*) as total_photos from photos) ,
5 cc as (select user_id, count(photo_id) as like_count from likes group by user_id)
6 select users.username, user_id, like_count from cc
7 inner join pp on cc.like_count = pp.total_photos
8 join users on cc.user_id=users.id ;
```

Result Grid:

username	user_id	like_count
Aniya_Hackett	5	257
Jadyn81	14	257
Roco33	21	257
Maxwell.Halvorson	24	257
Olie_Ledner37	36	257
MalikMunoz17	41	257

Action Output:

Time	Action	Message	Duration / Fetch
55 12:38:53	with pp as (select count(*) as total_photos from photos) , cc as (select user_id, count(photo_id) as like_count from likes group by user_id)	13 row(s) returned	0.000 sec / 0.000 sec
56 12:38:54	select users.username, user_id, like_count from cc inner join pp on cc.like_count = pp.total_photos	13 row(s) returned	0.000 sec / 0.000 sec
57 12:38:54	join users on cc.user_id=users.id ;	13 row(s) returned	0.000 sec / 0.000 sec