

Instagram User Analytics

Project Description: This project uses several database management technologies to extract valuable insights from the raw data, which will be then visualised to enhance platform functionality.

Approach: SQL was used to carry out this project, and queries were used to turn the given raw data into a database. The necessary data and insights were then obtained by applying sorting and data extraction queries.

Tech-Stack Used: The tech stack contains Mysql Workbench 8.0.37, a great tool for database querying because of its user-friendly interface, straightforward setup process, and support for troubleshooting.

Project insights:

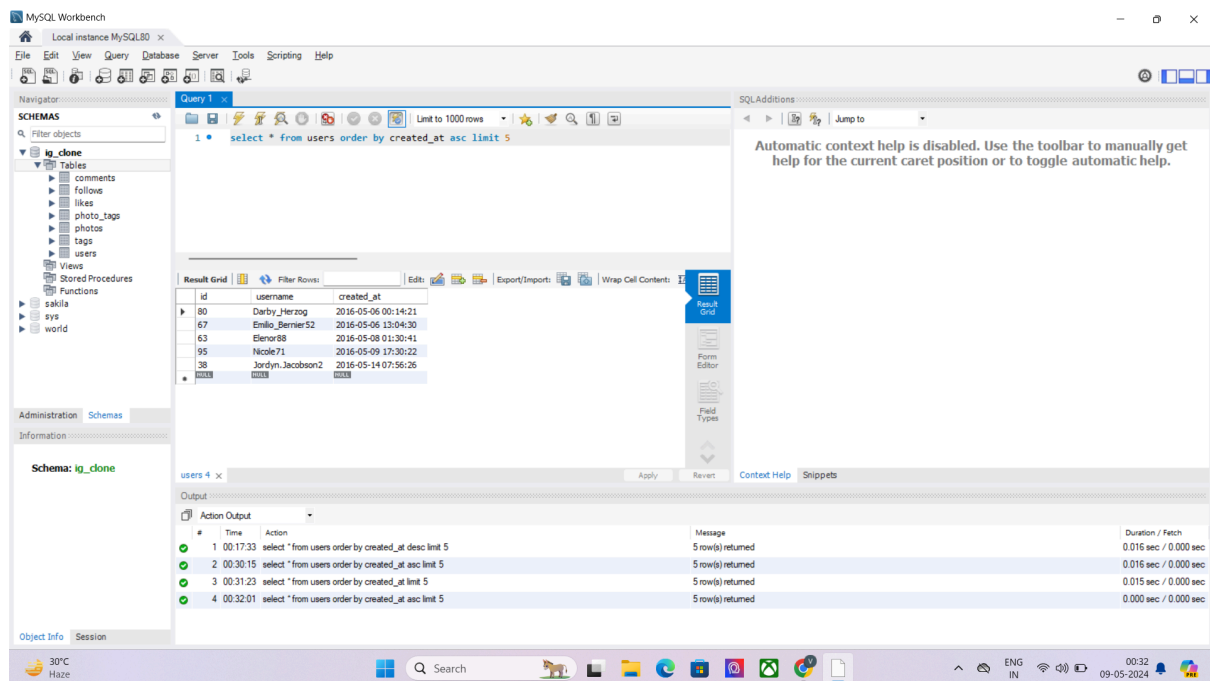
A) Marketing:

1. Reward Most Loyal Users: People who have been using Instagram for the longest time.

Result: The Five oldest users of Instagram.

Id	Username	Created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	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

Output Screenshot:



Code:

```
select * from users order by created_at asc limit 5
```

2. Inactive user Engagement: Remind inactive users to start posting by sending them promotional emails to post their first picture on the platform.

Result: The users who are inactive.

- 5 Aniya_Hackett
- 7 Kasandra_Homenick
- 14 Jaclyn81
- 21 Rocio33
- 24 Maxwell.Halvorson
- 25 Tierra.Trantow
- 34 Pearl7
- 36 Ollie_Ledner37
- 41 Mckenna17
- 45 David.Osinski47
- 49 Morgan.Kassulke
- 53 Linnea59

54 Duane60

57 Julien_Schmidt

66 Mike.Auer39

68 Franco_Keebler64

71 Nia_Haag

74 Hulda.Macejkovic

75 Leslie67

76 Janelle.Nikolaus81

80 Darby_Herzog

81 Esther.Zulauf61

83 Bartholome.Bernhard

89 Jessyca_West

90 Esmeralda.Mraz57

91 Bethany20

Output Screenshot:

The screenshot shows a database query interface. At the top, there are tabs for 'Query 1', 'ig_clone - Schema', 'users', 'ig_clone.likes', and 'follows'. Below the tabs is a toolbar with various icons, including a 'Limit to 1000 rows' dropdown. The SQL query is displayed in the main area:

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

Below the query, the 'Result Grid' is shown. It contains a table with two columns: 'id' and 'username'. The table lists 15 rows of data:

id	username
5	Aniya_Hackett
7	Kassandra_Homenick
14	Jadyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59

On the right side of the interface, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom, there is a 'Read Only' status indicator.

Code:

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

3. Contest Winner Declaration: The team started a contest and the user who gets the most likes on a single photo will be the winner and the team now wants to announce the winner.

Result: The user with most likes on their one photo.

Username	image_url	used_id	likes
Zack_Kemmer93	https://jarret.name	52	48

Code:

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

Output Screenshot:

Query 1 x ig_clone - Schema users ig_clone.likes follows

Limit to 1000 rows

```

1 • select users.username, photos.image_url, photos.user_id, count(likes.photo_id) likes from photos
2   join likes on likes.photo_id = photos.id
3   join users on users.id = photos.user_id
4   group by photos.image_url, photos.user_id
5   order by count(likes.photo_id) desc
6   limit 1;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

username	image_url	user_id	likes
Zack_Kemmer93	https://jarret.name	52	48

Result 40 x Read Only

4. Hashtag Research: A brand partner is interested in finding the most widely used hashtags to incorporate into their postings in order to maximise their audience reach.

Result: Here are the most Five trending hashtags which partner brand can use.

smile 59

beach 42

party 39

fun 38

concert 24

Output Screenshot:

The screenshot shows a database query editor interface. At the top, there are tabs for 'Query 1', 'ig_clone - Schema', 'users', 'ig_clone.likes', and 'follows'. Below the tabs is a toolbar with various icons. The main area contains a SQL query:

```
9 • select * from tags
10
11 select tag_name, count(tags.id) from tags
12 left join photo_tags on photo_tags.tag_id = tags.id
13 where photo_tags.tag_id is not null
14 group by tag_name
15 order by count(tags.id) desc limit 5
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button. The result grid displays the following data:

tag_name	count(tags.id)
concert	24
fun	38
party	39
beach	42
smile	59

On the right side of the interface, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'.

Code:

```
select tag_name, count(tags.id) from tags
left join photo_tags on photo_tags.tag_id = tags.id
where photo_tags.tag_id is not null
group by tag_name
order by count(tags.id) desc
limit 5
```

5. Ad Campaign Launch: The team wants to know which days would be the best days to launch the Campaign Ads.

Result: These are the best days to launch the Campaign Ads.

- 16 Thursday
- 16 Sunday
- 15 Friday
- 14 Tuesday
- 14 Monday
- 13 Wednesday

12 Saturday

Output Screenshot:

The screenshot shows a database query editor interface. At the top, there are tabs for 'Query 1', 'ig_clone - Schema', 'users', 'ig_clone.likes', and 'follows'. Below the tabs is a toolbar with various icons. The main area contains a SQL query:

```
16
17
18
19 select count(id)user_count,dayname(created_at) from users
20 group by dayname(created_at)
21 order by count(id) desc
22
23
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows:' input field, an 'Export:' button, and a 'Wrap Cell Content:' checkbox. The results are displayed in a table with two columns: 'user_count' and 'dayname(created_at)'. The table contains the following data:

	user_count	dayname(created_at)
▶	15	Friday
	14	Monday
	12	Saturday
	16	Sunday
	16	Thursday
	14	Tuesday
	13	Wednesday

On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'.

Code:

```
select count(id)user_count,dayname(created_at) from users
group by dayname(created_at)
order by count(id) desc
```

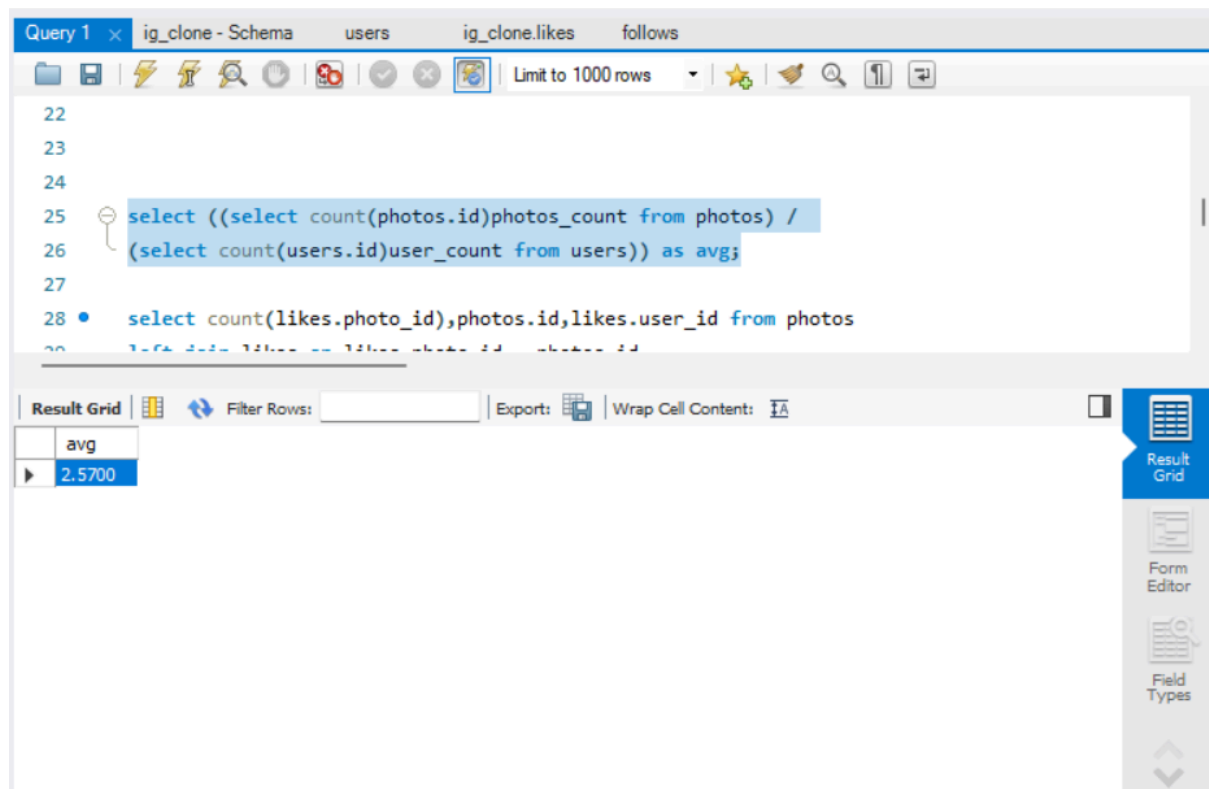
B) Investor Metrics

1. User Engagement: Investors are interested in knowing whether Instagram users are still posting regularly or if they are using it less frequently.

Result: A user average post is more than 2.

2.5700

Output Screenshot:



Code:

```

select ((select count(photos.id)photos_count from photos) /
(select count(users.id)user_count from users)) as avg;

```

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

Result: These are the users who can be bots and have fake accounts.

username	photo_likes	user_id
Aniya_Hackett	257	5
Jaclyn81	257	14
Rocio33	257	21
Maxwell.Halvorson	257	24
Ollie_Ledner37	257	36

Mckenna17	257	41
Duane60	257	54
Julien_Schmidt	257	57
Mike.Auer39	257	66
Nia_Haag	257	71
Leslie67	257	75
Janelle.Nikolaus81	257	76
Bethany20	257	91

Output Screenshot:

The screenshot shows a SQL query editor interface. The query is as follows:

```

37
38 select * from users
39
40 select * from (select users.username, count(likes.photo_id) photo_likes,
41 (case when count(likes.photo_id) = (select count(id) from photos)
42 then likes.user_id end) user_id from likes
43 join users on users.id = likes.user_id
44 group by likes.user_id, users.username)t
45 where user_id is not null

```

The results are displayed in a table with the following columns: username, photo_likes, and user_id.

username	photo_likes	user_id
JadyN81	257	14
Rocio33	257	21
Maxwell.Halvorson	257	24
Ollie_Ledner37	257	36
Mckenna17	257	41
Duane60	257	54
Julien_Schmidt	257	57
Mike.Auer39	257	66
Nia_Haag	257	71
Leslie67	257	75

The interface also includes a 'Result Grid' tab, a 'Filter Rows' section, and an 'Export' button. The bottom right corner shows a 'Read Only' status.