

Instagram User Analytics

Project Description :

The name of the project **Instagram User Analytics** it self contains the whole description of the project. In this valuable project we are going to analyse the data of application named Instagram. Particularly, focus on the data related to users of Instagram and make valuable decisions for future planning. There are several tables available in database ig_clone and we performed accurate queries on it and tried to give perfect answers of the task.

The insights derived from this analysis can be used by various teams within the business. This project will help specially the Marketing team, product team as well as development team to take the perfect decision to improve the business.

A) Marketing Analysis:

1. **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.

Query :

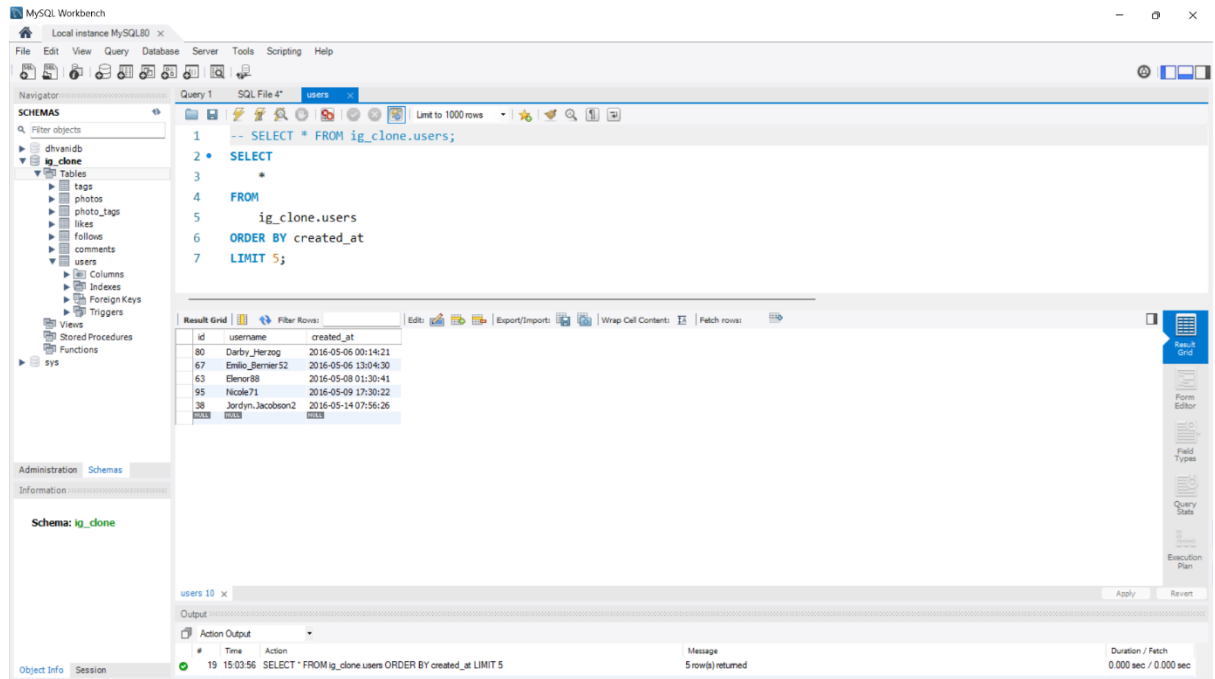
```
SELECT
*
FROM
ig_clone.users
ORDER BY created_at
LIMIT 5;
```

OUTPUT:

	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
*	NULL	NULL	NULL

Insight :

This are the tops five users who are using Instagram for the long time. Marketing team can give the rewards to this five oldest users on Instagram.



The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'dhvaniadb' and 'ig_clone' (selected). Under 'ig_clone', there are tables like 'tags', 'photo', 'photo_tags', 'likes', 'follows', 'comments', and 'users'. The 'users' table is selected. The main editor shows a SQL query:

```
1 -- SELECT * FROM ig_clone.users;  
2 *  
3  
4 FROM  
5     ig_clone.users  
6 ORDER BY created_at  
7 LIMIT 5;
```

The 'Result Grid' shows the following data:

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	Jordym.Jacobson2	2016-05-14 07:56:26

The 'Output' pane at the bottom shows the execution details:

#	Time	Action	Message	Duration / Fetch
19	15:03:56	SELECT * FROM ig_clone.users ORDER BY created_at LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec

2. **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:

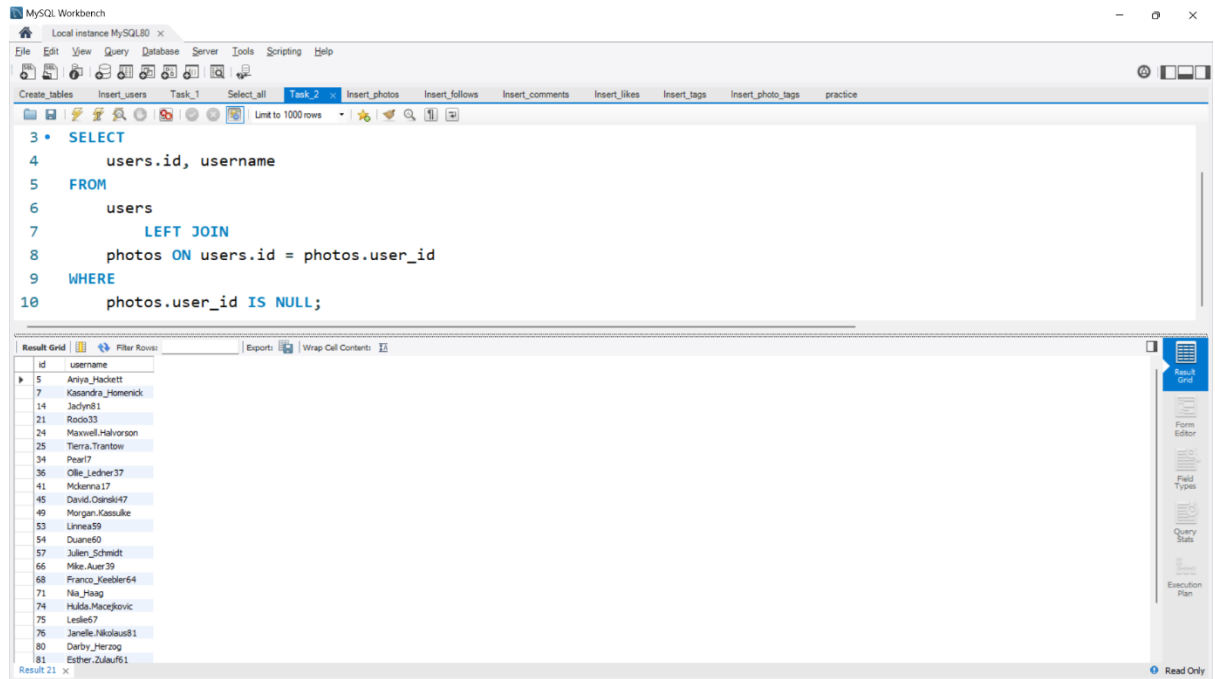
```
SELECT
    users.id, username
FROM
    users
    LEFT JOIN
    photos ON users.id = photos.user_id
WHERE
    photos.user_id IS NULL;
```

OUTPUT:

	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
	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

Insight :

From this, marketing team can decide that which users are not engaging on application. They can send the mails to encourage them to share their thoughts via posts.



The screenshot displays the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar contains icons for creating tables, inserting data, and running queries. The main query editor shows a SQL query that selects user IDs and usernames from the 'users' table, joined with the 'photos' table on the condition that the user ID is not null. The query is as follows:

```
3 • SELECT
4     users.id, username
5 FROM
6     users
7     LEFT JOIN
8     photos ON users.id = photos.user_id
9 WHERE
10    photos.user_id IS NULL;
```

Below the query editor, the 'Result Grid' tab is active, showing a table with two columns: 'id' and 'username'. The table contains 21 rows of data, with the first row highlighted. The results are as follows:

id	username
5	Aniya_Hackett
7	Kassandra_Homenick
14	Jacyln81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Tranter
34	Pearl7
36	Olle_Ledner37
41	McLenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
65	Mila_Auer29
68	Franco_Keebler64
71	Nia_Haag
74	Hilda.Macejkovic
75	Leslie57
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther_Zulauf1

The bottom status bar indicates 'Result 21' and 'Read Only'.

3. **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:

```

SELECT
    photos.user_id AS id_of_user,
    users.username AS name_of_user,
    COUNT(likes.photo_id) AS total_likes_on_post,
    likes.photo_id AS id_of_photo,
    photos.image_url AS photo_url
FROM
    photos
    RIGHT JOIN
    likes ON photos.id = likes.photo_id
    LEFT JOIN
    users ON photos.user_id = users.id
GROUP BY id_of_photo
ORDER BY total_likes_on_post DESC
LIMIT 1;

```

OUTPUT:

	id_of_user	name_of_user	total_likes_on_post	id_of_photo	photo_url
▶	52	Zack_Kemmer93	48	145	https://jarret.name

Insight:

Marketing team should give the price to the user whose user name is Zack_Kemmer93, which contains the highest likes amongst all which is around 48.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

5 • use ig_clone;
6
7 • SELECT
8     photos.user_id AS id_of_user,
9     users.username AS name_of_user,
10    COUNT(likes.photo_id) AS total_likes_on_post,
11    likes.photo_id AS id_of_photo,
12    photos.image_url AS photo_url
13 FROM
14     photos
15     RIGHT JOIN
16     likes ON photos.id = likes.photo_id
17     LEFT JOIN
18     users ON photos.user_id = users.id
19 GROUP BY id_of_photo
20 ORDER BY total_likes_on_post DESC
21 LIMIT 1;

```

The Results Grid at the bottom shows the output of the query:

id_of_user	name_of_user	total_likes_on_post	id_of_photo	photo_url
52	Zack_Kemmer93	48	145	https://jarret.name

4. **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.

QUERY:

```

SELECT
    photo_tags.tag_id,
    tags.tag_name,
    COUNT(photo_tags.tag_id) AS total_use_of_tag
FROM
    photo_tags
    LEFT JOIN
        tags ON photo_tags.tag_id = tags.id
GROUP BY photo_tags.tag_id
ORDER BY total_use_of_tag DESC
LIMIT 5;

```

OUTPUT:

	tag_id	tag_name	total_use_of_tag
▶	21	smile	59
	20	beach	42
	17	party	39
	13	fun	38
	18	concert	24

Insight :

There are five tags which can be suggested to users to use in their posts to reach the most people which are smile, beach, party, fun and concert.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```

1 • select * from photo_tags;
2 • select * from tags;
3
4 • SELECT
5     photo_tags.tag_id,
6     tags.tag_name,
7     COUNT(photo_tags.tag_id) AS total_use_of_tag
8 FROM
9     photo_tags
10    LEFT JOIN
11        tags ON photo_tags.tag_id = tags.id
12 GROUP BY photo_tags.tag_id
13 ORDER BY total_use_of_tag DESC
14 LIMIT 5;

```

The Results Grid at the bottom displays the output of the query:

tag_id	tag_name	total_use_of_tag
21	smile	59
20	beach	42
17	party	39
13	fun	38
18	concert	24

5. **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.

QUERY:

```
SELECT
    DAYOFWEEK(created_at) AS day_of_week,
    DAYNAME(created_at) AS name_of_day,
    COUNT(DAYNAME(created_at)) AS total_register_user
FROM
    users
GROUP BY name_of_day , day_of_week
ORDER BY total_register_user DESC;
```

OUTPUT:

	day_of_week	name_of_day	total_register_user
▶	5	Thursday	16
	1	Sunday	16
	6	Friday	15
	3	Tuesday	14
	2	Monday	14
	4	Wednesday	13
	7	Saturday	12

Insight :

Marketing team can launch the new campaign on Thursday or Sunday because it can be shown that in this days most user registered on Instagram.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 • select * from users;
2
3 • SELECT
4     DAYOFWEEK(created_at) AS day_of_week,
5     DAYNAME(created_at) AS name_of_day,
6     COUNT(DAYNAME(created_at)) AS total_register_user
7 FROM
8     users
9 GROUP BY name_of_day , day_of_week
10 ORDER BY total_register_user DESC;
```

The output is displayed in the 'Result Grid' tab, showing the same data as the table above:

day_of_week	name_of_day	total_register_user
5	Thursday	16
1	Sunday	16
6	Friday	15
3	Tuesday	14
2	Monday	14
4	Wednesday	13
7	Saturday	12

The left sidebar shows the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', and 'users'. The 'users' table is selected, and its structure is shown in the 'Table: users' panel below:

```
Columns:
id          int(11) PK
username    varchar(255)
created_at  timestamp
```

B) Investor Metrics:

1. **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.

QUERY:

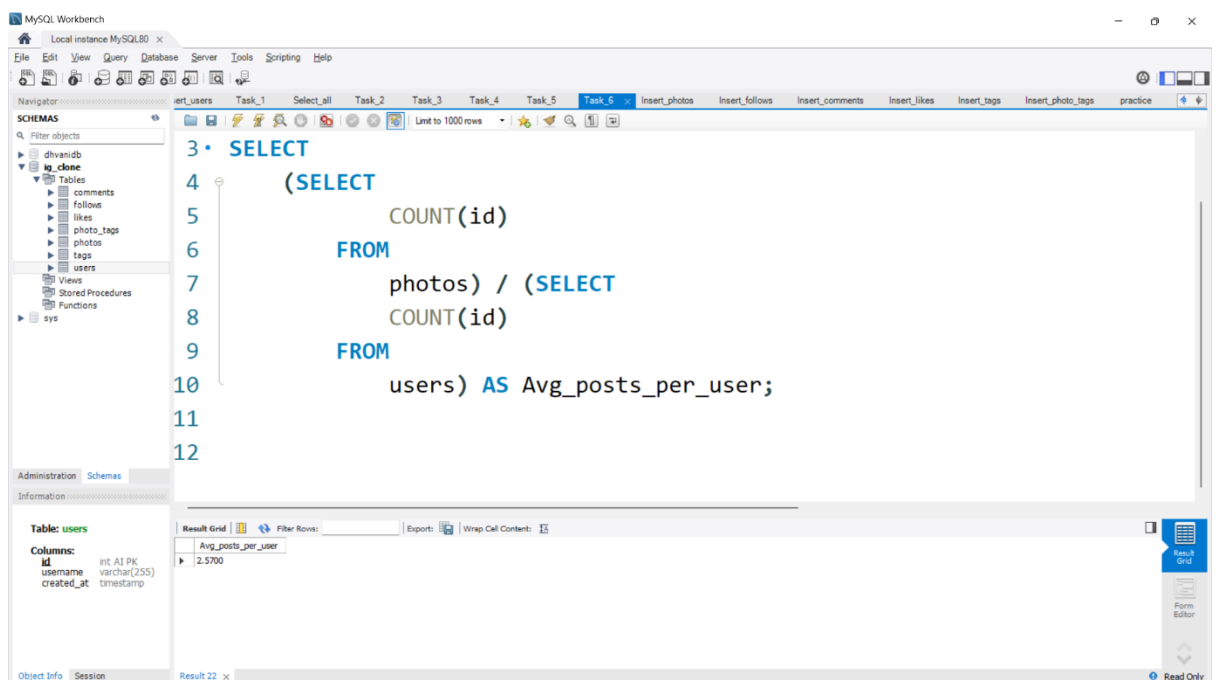
```
SELECT
  (SELECT
    COUNT(id)
  FROM
    photos) / (SELECT
    COUNT(id)
  FROM
    users) AS Avg_posts_per_user;
```

OUTPUT:

Avg_posts_per_user
2.5700

Insight :

From this, It can be said that every user post atleast around 3 posts in their account.



- Also, provide the total number of photos on Instagram divided by the total number of users.

QUERY:

```
SELECT
  COUNT(id)
FROM
  users;
SELECT
```



```
    COUNT(id)
FROM
  photos;SELECT
  (SELECT
    COUNT(id)
  FROM
    photos) / (SELECT
    COUNT(id)
  FROM
    users) AS photos_divided_by_users;
```

OUTPUT:

photos_divided_by_users
2.5700

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL query with line numbers 1 through 11. The query is as follows:

```
1 select count(id) from users;
2 select count(id) from photos;
3 SELECT
4     (SELECT
5         COUNT(id)
6     FROM
7         photos) / (SELECT
8         COUNT(id)
9     FROM
10        users) AS photos_divided_by_users;
```

The left sidebar shows the 'SCHEMAS' panel with a tree view of the database structure. The 'users' table is selected, and its columns are listed: 'id' (int AI PK), 'username' (varchar(255)), and 'created_at' (timestamp).

The bottom right corner shows the 'Result Grid' with the following output:

photos_divided_by_users
2.5700

2. **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.

QUERY:

```
SELECT
    likes.user_id,
    users.username,
    COUNT(likes.user_id) AS total_likes
FROM
    likes
    INNER JOIN
    users ON likes.user_id = users.id
GROUP BY likes.user_id
HAVING total_likes = (SELECT
    COUNT(id)
    FROM
    photos)
ORDER BY total_likes DESC;
```

OUTPUT:

	user_id	username	total_likes
►	5	Aniya_Hackett	257
	14	Jadyn81	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

Insight :

This are the users which can be fake and can misuse the application, so team can analyse this accounts and delete this accounts to save the privacy of the other users.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'dhvanidb' selected, showing tables like 'comments', 'follows', 'likes', 'photo_tags', 'tags', 'users', 'Views', 'Stored Procedures', and 'Functions'. The main editor window contains a SQL query in 'Task_7'.

```
1 • SELECT
2   likes.user_id,
3   users.username,
4   COUNT(likes.user_id) AS total_likes
5 FROM
6   likes
7   INNER JOIN
8   users ON likes.user_id = users.id
9 GROUP BY likes.user_id
10 HAVING total_likes = (SELECT
11   COUNT(id)
12 FROM
13   photos)
14 ORDER BY total_likes DESC;
```

The 'Result Grid' at the bottom shows the results of the query, limited to 30 rows. The columns are 'user_id', 'username', and 'total_likes'. The results are sorted by 'total_likes' in descending order.

user_id	username	total_likes
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell_Halvorson	257
36	Olle_Jedner37	257
41	Mickenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mica_Auer39	257
71	Nia_Heag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

The bottom left pane shows the 'Table: users' structure with columns: 'id' (int AI PK), 'username' (varchar(255)), and 'created_at' (timestamp).

Approach:

- The first step that I took towards this project was, I understand the connection between different tables and its data. Then, try to understand how I can make insights from the given data.
- Then after, I started from first query and understood the question, perform a step-by-step process to reach the perfect output.
- Finally, make some insights out of the derived output.

Tech-Stack Used:

In this project, I specifically used the **MYSQL Workbench 8.0 CE** to perform the required queries.

There are several **reasons** why I used such software:

1. MySQL is a specific type of SQL database management system.
2. MYSQL is easy to understand and provide easy way to perform the require queries.
3. MySQL is a relational database management system (RDBMS) that uses SQL as the standard query language, it's known for its scalability and open-source availability.

Result :

- I learnt the valuable things from this projects that how to perform queries, like I have to understand what I have to derive from the data and then perform query step-by-step.
- Moreover, I understand how this insights can help various teams in various field to make decisions and how this decision can help to grow the business.
- Last but not least, I learnt how to perform and how to interact with multiple tables and take desired outcomes from the data.
- This project give me the basic ideas how to work with the data and how to cross check the output.