INSTAGRAM USER ANALYTICS

OVERVIEW OF THE PROJECT:

PROJECT DESCRIPTION:

As a data analyst working with the product team at Instagram. my role involves analyzing user interactions and engagement with the Instagram app to provide valuable insights that can help the business grow. I have used SQL(majorly DML)commands to analyse database and extract information to get the desired output.

APPROACH :

Since the dataset was already given I've tried to analyse by using various commands executed using MYSQL workbench

I have mentioned the steps followed for each query below along with code and results

TECH STACK USED:

I have used MYSQL WORKBENCH 8.0 CE because this software is very easy to approach and queries can be easily executed

INSIGHTS :

I have personally gained lot of knowledge as I went on analysing data and tried each and every command carefully and executed, luckily through various learning mysql commands I have gotten an idea to use what to where followed by practising small databases and executing them

Once I have understood how to use appropriate commands and where to use it has been easy to analyse data and execute them and commands varies for every query it needs understanding technically and logically and once I have started solving every query it boosted my confidence to solve level by level and it made interesting to learn MYSQL

A) Marketing Analysis:

1. Loyal User Reward:

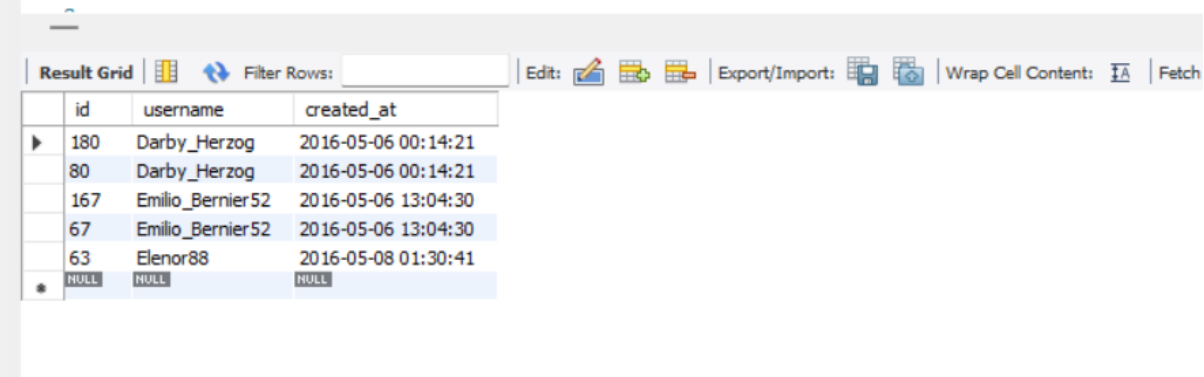
Description:

Here I have sorted created at coloumn and taken the top 5 members who are considered as loyal users and rewarded them

Code:

```
SELECT * from users
ORDER BY created_at
LIMIT 5 ;
```

Result:



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch

id	username	created_at
180	Darby_Herzog	2016-05-06 00:14:21
80	Darby_Herzog	2016-05-06 00:14:21
167	Emilio_Bernier52	2016-05-06 13:04:30
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
NULL	NULL	NULL

2.Inactive user engagement:

Description:

Here I have used left join command and found out the users whose photos are null and they are considered as inactive users

Code:

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

Result:

username
Aniya_Hackett
Kasandra_Homenick
Jaclyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59

3.contest winner declaration:

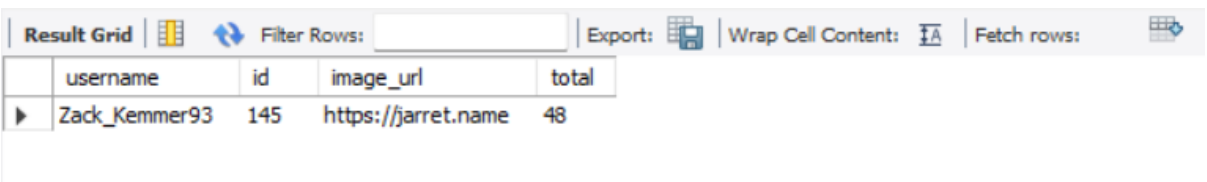
Description:

Here I have used inner join ,group by used to help with data combining, order by to sort result either in ascending or descending order and limited to 1 as the contest winner could possibly be the only person

Code:

```
SELECT username,photos.id,photos.image_url,count(likes.user_id) AS total
FROM photos
INNER JOIN likes
ON likes.photo_id=photos.id
INNER JOIN users
ON photos.user_id=users.id
GROUP BY photos.id
ORDER BY total DESC
LIMIT 1;
```

Result:



4.Hastag Research :

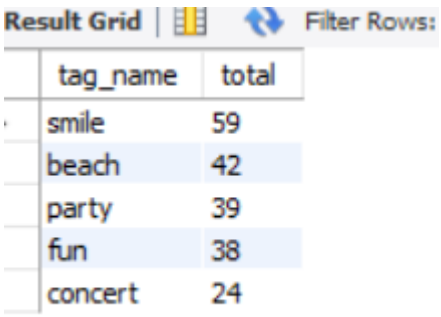
Description:

Here I have used join ,group by &order by commands and DESC to sort the data returned in descending order and limited to 5 popular most commonly hastags used

Code:

```
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;
```

RESULT;


Result Grid | Filter Rows:

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

5.Ad Campaign Launch:

Description:

Here we need to analyse best day of the week to launch ads so that most users would spend time on app and can literally go through ads we can use group by ,order by to extract information and I have limited to 3 best days to have the more user engagement

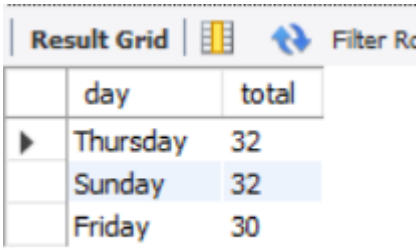
Code:

```
SELECT DAYNAME(Created_at) AS day,count(*) as total

FROM users

GROUP BY day
```

ORDER BY total DESC

LIMIT 3;

Result:



	day	total
▶	Thursday	32
	Sunday	32
	Friday	30

B.INVESTOR METRICS:

1.User Engagement:

Description:

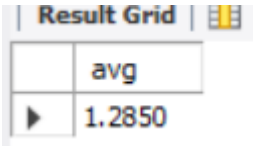
Here investors want to know if users are still active and posting on Instagram or if they are making fewer posts, to know by calculating average the total number of photos on Instagram divided by the total number of users.

Code:

SELECT

(SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS avg;

Result:



	avg
▶	1.2850

2.Bots &Fake Accounts:

Description:

Here Investors want to know if the platform is crowded with fake and dummy accounts if so then we can analyse users liking every single pictures present in the application by using group by , having clause to filter the result of group by query based on result of combining and also join command to find out all the bots and fake accounts

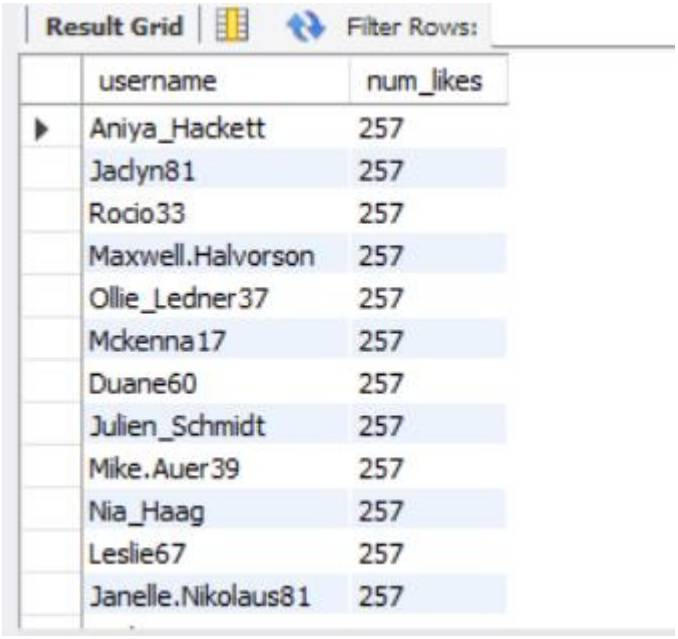
Code:

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);
```

Results:



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
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257