REPORT

Project Description:

The project is about analysis of Instagram User. Using these one can analyse how many users are engaged on these platform and how they are interacting with it. From these analysis we can easily find out things or insights which will help management team for various campaign like on which day users are active/ registered, how many posts they are sharing, likes, comments and many more.

For implementing these project I have used SQL as using this tech we can easily interact with data or multiple databases at once.

Approach :

I have used SQL for fetching the data results / insights as it is most efficient way to interact with data. For implementing these project first we need to learn what should be done to get desired output / insights. By using SQL queries I easily got the output as it is already tuned and once the query is written we can reuse it multiple times.

Tech-Stack Used:

I used XAMP server as in these software we can easily view the structure of the stored databases. And also it provides graphical representation of stored data.

Version : v3.3.0

Insights:

While implementing these project I came to know where SQL's Command such as DDL, DML. I got to learn various SQL's aggregate functions such as sum(), avg(), count() and so on. Using these one can derive descriptive statistics.

A) Marketing

1. Rewarding Most Loyal Users:

List of users who have been using the platform for the longest time:

Query:

Here I have used select statement to fetch the id and username of users, from keyword to specify from which table we want to fetched the data, order by keyword to sort data in ascending order and limit keyword to only fetch first 5 data

select users.id, users.username from users order by users.created_at limit 5;

Output:

a) Darby_Herzong (id: 80)
b) Emilio_Bernier52 (id: 67)
c) Elenor88 (id: 63)

d) Nicole71 (id: 95)
e) Jordyn.Jacobson2 (id: 38)

2. **Remind Inactive Users to Start Posting:**
List of users who have never posted a single photo on Instagram:

Query:

Here I have used select statement to fetch the id and username of users, from keyword to specify from which table we want to fetched the data, where clause to filter the data, Not In keywords to check which id are not present in photo table and subquery for selecting user_id from photos as it is a foreign key which is used to set a relation between two tables.Group By to group user ids having same values.

select users.id, users.username from users where users.id NOT IN (select photos.user_id from photos GROUP By photos.user_id);

Output:
a) Aniya_Hackett (id: 5)
b) Kasandra_Homenick (id: 7)
c) Jaclyn81 (id: 14)
d) Rocio33 (id: 21)
e) Maxwell.Halvorson (id: 24)
f) Tierra.Trantow (id: 25)
g) Pearl7 (id: 34)
h) Ollie_Ledner37 (id: 36)
i) Mckenna17 (id: 41)
j) David.Osinski47 (id: 45)
k) Morgan.Kassulke (id: 49)
l) Linnea59 (id: 53)
m) Duane60 (id: 54)
n) Julien_Schmidt (id: 57)
o) Mike.Auer39 (id: 66)
p) Franco_Keebler64 (id: 68)
q) Nia_Haag (id: 71)
r) Hulda.Macejkovic (id: 74)
s) Leslie67 (id: 75)
t) Janelle.Nikolaus81 (id: 76)
u) Darby_Herzog (id: 80)
v) Esther.Zulauf61 (id: 81)
w) Bartholome.Bernhard (id: 83)
x) Jessyca_West (id: 89)
y) Esmeralda.Mraz57 (id: 90)
z) Bethany20 (id: 91)

3. **Declaring Contest Winner**:
The details of winner of contest:

Query:

Here I have used select statement to fetch the username of users, id of photo, URL of photo that I got most likes and to fetch output of subqueries. I have used from keyword to specify from which table we want to fetched the data. After that I have used Inner join to join and fetch the particular data from likes table as Inner Join help us to find out the records that are present in both the table. As clause is used to give a name of column and to specify a result of subqueries with the particular name. I have used aggregate function count to count the maximum likes got for particular photo. Order By function to sort the count of like and limit is used to fetch one single record

SELECT users.username, photos.id, photos.image_url, most_likes.like_count FROM photos INNER JOIN (SELECT photo_id, COUNT(*) AS like_ count FROM likes GROUP BY photo_id) AS most_likes INNER JOIN users ON users.id = photos.user_id ORDER BY like_count DESC LIMIT 1;

Output:

Name: Kenton_Kirlin
Id: 1
Likes: 48
Photo URL: http://elijah.biz

4. **Hashtag Researching:**

List of 5 most commonly used hashtags on the platform:

Query:

Here I have used select statement to fetch the tag id, name of tag and aggregate function count to get count of tag name. I have used As clause to give a name to count column. Inner Join on photo_tags to get record that are present in both table i.e, photo_tags and tags. I have used Group By Clause to group tag ids having same values , Order by to get total in descending order and Limit to fetch first 5 records.

SELECT tags.id, tags.tag_name, COUNT(tags.tag_name) AS total FROM tags INNER JOIN photo_tags ON tags.id = photo_tags.tag_id GROUP BY tags.id ORDER BY total DES C LIMIT 5;

Output:

a) Smile (id: 21, total: 59)
b) Beach (id: 20 total: 42)
c) Party (id: 17 total: 39)
d) fun (id: 13 total: 38)

e) concert (id: 18 total: 24)

5. **Launch AD Campaign:**

Day of the week on which most users are registered.

Query:

Here I have used select statement to name of the day for which I have used DAYNAME() function which will return day from date, and aggregate function count to get count of total records present in users. As clause to fetch a column by the specified name. From keyword to specify the name of table from which the data should get fetched. Group By clause to group the data / rows that has same values. Order By clause to sort the count in descending order.

SELECT DAYNAME(created_at) AS day,COUNT(*) AS total_registrations FROM users GR OUP BY day ORDER BY total_registrations DESC;

Output:

a) Sunday (Count: 16).
b) Thursday (Count : 16).
c) Friday (Count: 15).
d) Tuesday (Count : 14).
e) Monday (Count: 14).
f) Wednesday (Count : 13).
g) Saturday (Count : 12).

As per the result ad campaign should schedule either on Sunday or Thursday.

B) Investor Metrics:

1. **User Engagement:**

Query:

Here I have used select statement to fetch the data. Using concept of subqueries I have calculated the total count by using count aggregate function and then divided both the total.

select ((select count(*) from photos) / (select count(*) from users));

Output:

Average user post on Instagram for 2.57 time.
the total number of photos on Instagram/total number of users: 257.

2. Bots & Fake Accounts:

users (bots) who have liked every single photo on the site:

Query:

I have used select statement to fetch the id of users, name of users and count of user id. As clause to fetch total count by specified name. Inner join on like table to fetch the data which is present in both table I.e, users and likes. Group By to fetch the row that has same value and Having clause to check the condition as we cannot use where condition to aggregate function.

```
SELECT users.id,users.username, COUNT(users.id) As total_like_count FROM users INN
ER
JOIN likes ON users.id = likes.user_id GROUP BY users.id HAVING total_like_count= (SE
LECT COUNT(*) FROM photos);
```

Output:

a) Aniya_Hackett (id:5 , count: 257).
b) aclyn81 (id: 14, count: 257).
c) Rocio33 (id: 21, count: 257).
d) Maxwell.Halvorson (id: 24, count: 257).
e) Ollie_Ledner37 (id: 36, count: 257).
f) Mckenna17 (id: 41, count: 257).
g) Duane60 (id: 54, count: 257).
h) Julien_Schmidt (id: 57, count: 257).
i) Mike.Auer39 (id: 66, count: 257).
j) Nia_Haag (id: 71, count: 257).
k) Leslie67 (id: 75, count: 257).
l) Janelle.Nikolaus81 (id: 76, count: 257).
m) Bethany20 (id: 91, count: 257).

Result:

As a result I got a clean, specific data which can be used to analyse various factors of Instagram. These analysis helped to understand how we can used SQL for data analysis and what is the importance of gathering, cleaning and storing the data.

Drive Link :

https://drive.google.com/file/d/1FZQB1AJF4CCI85cXy2HTVgi9khGEA033/view?usp=share_link