Instagram User Analytic

A) **Project Description:**This project aims to extract useful insights from raw data/metadata, using various database management tools, and even visualize them to increase the platform's efficiency.
B) **Project Approach:**The project was executed using SQL, where queries were utilized to create a database from the provided raw data. Sorting and data extracting queries were then implemented to obtain the required data/insights.
C) **Tech Stack Used**:The tech stack used included MySQL Workbench v8.0.30.0, which was an excellent tool for querying the database, thanks to its ease of access, simple setup, and GUI, as well as its troubleshooting support.

Project Insights:(Raw Insights :)A) Marketing:

1. Rewarding Most Loyal Users: People who have been using the platform for the longest time.

Conclusion:These are the oldest user of instagram.

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

Code: SELECT *

FROM users

ORDER BY created_at

LIMIT 5;

2. Remind Inactive Users to Start Posting: By sending them promotional emails to post their 1st photo.

Conclusion:These users were inactive after their first post.

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

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

3. Declaring Contest Winner: The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner.
Conclusion:He has the most likes in his one post.
Zack_Kemmer93 145 https://jarret.name 48

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;
```

4. Hashtag Researching: A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.
Conclusion: These are some trending hastags which a partner brand can use.

smile	59
beach	42
party	39
fun	38
food	24
lol	24
concert	24
hair	23
happy	22
beauty	20
dreamy	20
sunset	19
fashion	19
drunk	19
sunrise	17
landscape	17
style	17
photography	16
stunning	16
delicious	15
foodie	11

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;
```

```
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;
```

5. Launch AD Campaign: The team wants to know, which day would be the best day to launch ADs.

Conclusion: these days would be best for AD campaign.

Thursday 16
Sunday 16

Code:

```
SELECT
    DAYNAME(created_at) AS day,
    count(*) as total
FROM users
GROUP BY day
ORDER BY total DESC
LIMIT 2;
```

B)Investor Metrics

6. User Engagement: Are users still as active and post on Instagram or they are making fewer posts
Conclusion:A users avaerage post is more than 2.

2.5700

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

7. Bots & Fake Accounts: The investors want to know if the platform is crowded with fake and dummy accounts
Conclusion:These are some user who can be boat and fake account.

Aniya_Hackett	257
Bethany20	257
Duane60	257
Jaclyn81	257
Janelle.Nikolaus81	257
Julien_Schmidt	257
Leslie67	257
Maxwell.Halvorson	257
Mckenna17	257
Mike.Auer39	257
Nia_Haag	257
Ollie_Ledner37	257
Rocio33	257

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);