Instagram User Analytics

A. Marketing Analysis:

Task 1: Identify the five oldest users on Instagram from the provided database.

SELECT *

FROM

users

ORDER BY created_at

LIMIT 5;

```
# Task 1 Loyal User Reward: Identify the five oldest users on Instagram from the provided database.
  4
  5 •
        SELECT
        FROM
            users
        ORDER BY created_at
 10
        LIMIT 5;
                                                                                                           | Edit: 🚄 📆 🖺 | Export/Import: 📳 🐻 | Wrap Cell Content: 🏗 | Fetch rows:
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
                 2016-05-09 17:30:22
  95 Nicole71
        Jordyn. Jacobson 2 2016-05-14 07:56:26
  38
  NULL
```

Task 2: Identify users who have never posted a single photo on Instagram.

SELECT

FROM

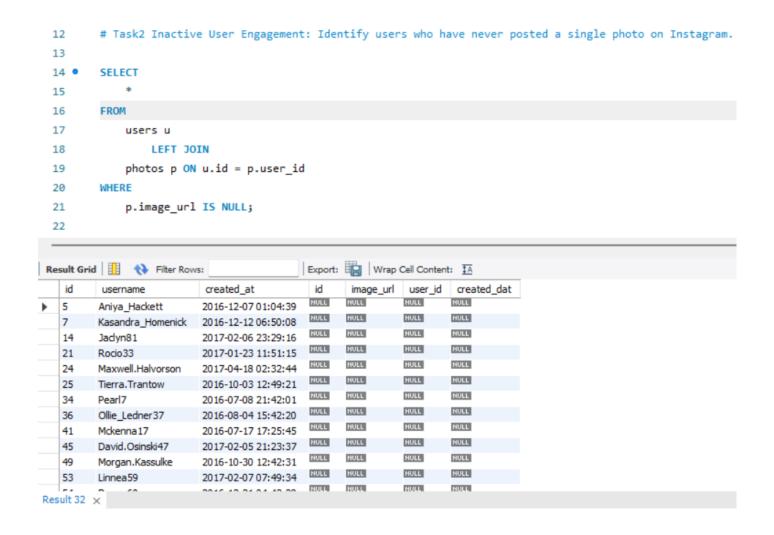
users u

LEFT JOIN

photos p ON u.id = p.user_id

WHERE

p.image_url IS NULL;



Task 3: Determine the winner of the contest and provide their details to the team.

```
SELECT
```

```
u.username,

l.photo_id,

p.image_url,

COUNT(l.user_id) AS Likess

FROM

likes I

INNER JOIN

photos p ON p.id = l.photo_id

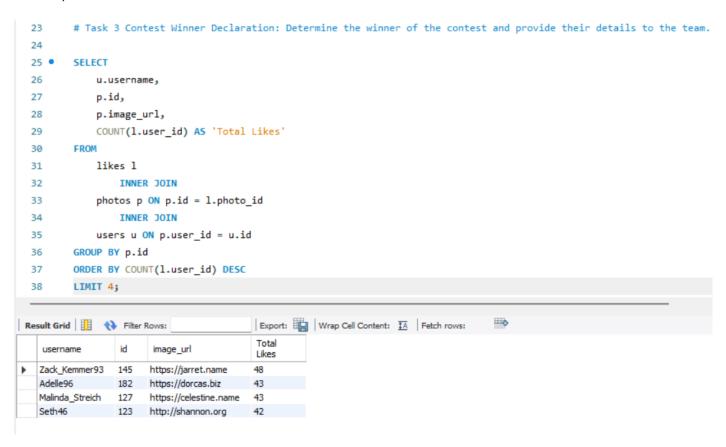
INNER JOIN

users u ON p.user_id = u.id

GROUP BY l.photo_id

ORDER BY Likess DESC
```

LIMIT 5;



Task 4: Identify and suggest the top five most commonly used hashtags on the platform.

SELECT

LIMIT 5;

```
t.tag_name, COUNT(pt.photo_id) AS Total_tags

FROM

photo_tags pt

INNER JOIN

tags t ON pt.tag_id = t.id

GROUP BY t.tag_name

ORDER BY Total_tags DESC
```

```
40
        # Task 4 Hashtag Research: Identify and suggest the top five most commonly used hashtags on the platform.
 41
 42 •
        select t.tag_name, count(*) as Total_tags from photo_tags pt
 43
        inner join tags t
        on pt.tag_id = t.id
 44
        group by t.id
 45
        order by Total_tags desc
 46
 47
        limit 5;
 48
Export: Wrap Cell Content: TA Fetch rows:
   tag_name Total_tags
  smile
           59
           42
  beach
           39
  party
  fun
           38
  food
           24
```

Task 5: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

SELECT

DAYNAME(created_at) AS Day,

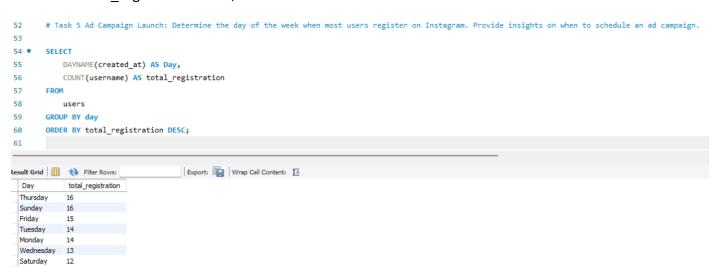
COUNT(username) AS total_registration

FROM

users

GROUP BY day

ORDER BY total registration DESC;



B. Investor Metrics:

Task 6 User Engagement: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

```
SELECT
 AVG(p.id) AS avg_number_of_posts
FROM
  photos p;
SELECT
 AVG(u.id) AS avg_number_of_users
FROM
  users u;
SELECT
  ROUND((SELECT
          AVG(p.id) AS avg_number_of_posts
        FROM
          photos p) / (SELECT
          AVG(u.id) AS avg_number_of_users
        FROM
          users u),
      1) AS avg_number_of_posts_per_user;
SELECT
  SUM(p.id) AS total_number_of_posts
FROM
  photos p;
```

SELECT

```
SUM(u.id) AS total_number_of_users

FROM

users u;

SELECT

ROUND((SELECT

SUM(p.id) AS total_number_of_posts

FROM

photos p) / (SELECT

SUM(u.id) AS total_number_of_users

FROM

users u),

1) AS total_number_of_posts_per_user;
```

```
# Task 6 User Engagement: Calculate the average number of posts per user on Instagram.
 62
        # Also, provide the total number of photos on Instagram divided by the total number of users.
 63
 64
 65 •
        SELECT
            AVG(p.id) AS avg_number_of_posts
 66
        FROM
 67
 68
            photos p;
 69
        SELECT
 70 •
 71
            AVG(u.id) AS avg_number_of_users
        FROM
 72
 73
            users u;
 74
        SELECT
 75 •
            ROUND((SELECT
 76
                             AVG(p.id) AS avg_number_of_posts
 77
                         FROM
 78
 79
                             photos p) / (SELECT
                             AVG(u.id) AS avg_number_of_users
 80
 81
                         FROM
 82
                             users u),

    AS avg_number_of_posts_per_user;

 83
Export: Wrap Cell Content: $\overline{A}$
   avg_number_of_posts_per_user
 2.6
```

```
85 •
        SELECT
            SUM(p.id) AS total number of posts
 86
        FROM
 87
            photos p;
 88
 89
 90 •
        SELECT
            SUM(u.id) AS total_number_of_users
 91
 92
        FROM
 93
            users u;
 94
 95 •
        SELECT
            ROUND ((SELECT
 96
 97
                             SUM(p.id) AS total_number_of_posts
 98
                         FROM
                             photos p) / (SELECT
 99
                             SUM(u.id) AS total number of users
100
101
                         FROM
                             users u),
102

    AS total_number_of_posts_per_user;

103
104
105
                                         Export: Wrap Cell Content: TA
total_number_of_posts_per_user
  6.6
```

Task 7 Bots & Fake Accounts: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

SELECT

```
u.id, u.username, COUNT(u.username) AS Total_likes
```

FROM

users u

JOIN

likes I ON l.user_id = u.id

```
GROUP BY u.id

HAVING Total_likes = (SELECT

COUNT(*)

FROM
```

photos p);

```
# Task 7 Bots & Fake Accounts: Identify users (potential bots) who have liked every single photo on the site,
 105
         # as this is not typically possible for a normal user.
106
107
         SELECT
108 •
             u.id, u.username, COUNT(u.username) AS Total_likes
109
110
111
             users u
                 JOIN
112
113
            likes 1 ON l.user_id = u.id
         GROUP BY u.id
114

→ HAVING Total_likes = (SELECT)

115
116
                 COUNT(*)
117
             FROM
118
                 photos p);
                                        Export: Wrap Cell Content: IA
Total_likes
         username
   36
         Ollie_Ledner37
   41 Mckenna 17
                     257
   54
        Duane60
                        257
   57 Julien_Schmidt 257
   66
        Mike.Auer39
   71 Nia_Haag
                       257
   75
                        257
        Leslie67
   76
        Janelle Nikolaus 81 257
        Bethany20
Result 56 ×
```