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;
```

```
3     # Task 1 Loyal User Reward: Identify the five oldest users on Instagram from the provided database.
4
5 •   SELECT
6         *
7     FROM
8         users
9     ORDER BY created_at
10    LIMIT 5;
```

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

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

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;
```

```sql
# Task2 Inactive User Engagement: 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	username	created_at	id	image_url	user_id	created_dat
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
14	Jadyn81	2017-02-06 23:29:16	NULL	NULL	NULL	NULL
21	Rocio33	2017-01-23 11:51:15	NULL	NULL	NULL	NULL
24	Maxwell.Halvorson	2017-04-18 02:32:44	NULL	NULL	NULL	NULL
25	Tierra.Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL
34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL
36	Ollie_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
41	Mckenna17	2016-07-17 17:25:45	NULL	NULL	NULL	NULL
45	David.Osinski47	2017-02-05 21:23:37	NULL	NULL	NULL	NULL
49	Morgan.Kassulke	2016-10-30 12:42:31	NULL	NULL	NULL	NULL
53	Linnea59	2017-02-07 07:49:34	NULL	NULL	NULL	NULL

Result 32

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

```sql
SELECT
   u.username,
   l.photo_id,
   p.image_url,
   COUNT(l.user_id) AS Likess
FROM
   likes l
      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;

```
23    # Task 3 Contest Winner Declaration: Determine the winner of the contest and provide their details to the team.
24
25    SELECT
26        u.username,
27        p.id,
28        p.image_url,
29        COUNT(l.user_id) AS 'Total Likes'
30    FROM
31        likes l
32            INNER JOIN
33        photos p ON p.id = l.photo_id
34            INNER JOIN
35        users u ON p.user_id = u.id
36    GROUP BY p.id
37    ORDER BY COUNT(l.user_id) DESC
38    LIMIT 4;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

username	id	image_url	Total Likes
Zack_Kemmer93	145	https://jarret.name	48
Adelle96	182	https://dorcas.biz	43
Malinda_Streich	127	https://celestine.name	43
Seth46	123	http://shannon.org	42

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

```
SELECT
   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
LIMIT 5;
```

```sql
# Task 4 Hashtag Research: Identify and suggest the top five most commonly used hashtags on the platform.

select t.tag_name, count(*) as Total_tags from photo_tags pt
inner join tags t
on pt.tag_id = t.id
group by t.id
order by Total_tags desc
limit 5;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

tag_name	Total_tags
smile	59
beach	42
party	39
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;

```sql
# Task 5 Ad Campaign Launch: 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;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Day	total_registration
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

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;
```

```sql
# 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_number_of_posts_per_user
2.6

```sql
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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 l ON l.user_id = u.id
```

```
GROUP BY u.id
HAVING Total_likes = (SELECT
    COUNT(*)
  FROM
    photos p);
```

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	username	Total_likes
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

Result 56