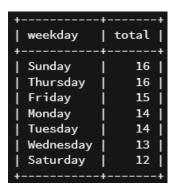
IG clone exercises solutions

1. Find the 5 oldest users.

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

++	+ 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

2. What day of the week do most users register on?



3. Find the users who have never posted a photo

```
SELECT

users.id,

username

FROM users

LEFT JOIN photos

ON users.id = photos.user_id

WHERE

photos.id IS NULL
;
```

```
| id | username
 5 | Aniya_Hackett
      Bartholome.Bernhard
      Bethany20
 80 |
      Darby_Herzog
 45
      David.Osinski47
      Duane60
 54
      Esmeralda.Mraz57
 90
 81
      Esther.Zulauf61
 68 | Franco_Keebler64
 74 | Hulda.Macejkovic
14 | Jaclyn81
      Janelle.Nikolaus81
 76
 89 |
      Jessyca_West
      Julien_Schmidt
      Kasandra_Homenick
 75
      Leslie67
      Linnea59
 53
      Maxwell.Halvorson
 24
      Mckenna17
      Mike.Auer39
 66
      Morgan.Kassulke
 49
      Nia_Haag
 71
 36 |
      Ollie_Ledner37
 34
      Pearl7
 21
      Rocio33
      Tierra.Trantow
```

4. Who uploaded the most liked photo?

```
SELECT
    users.username,
    photos.id as pShoto_id,
    COUNT(*) AS total_likes
FROM photos
JOIN likes
    ON likes.photo_id = photos.id
JOIN users
    ON users.id = photos.user_id
GROUP BY photos.id
ORDER BY total_likes DESC
LIMIT 1
;
```

5. How many times does the average user post?

My solution:

```
SELECT

AVG(n_photos) AS average

FROM(

SELECT

CASE

WHEN photos.id IS NULL THEN 0

ELSE COUNT(*)

END AS n_photos

FROM users

LEFT JOIN photos

ON users.id = photos.user_id

GROUP BY users.id

) AS photos_by_users;
```

Instructor's solution:

```
SELECT (SELECT Count(*) FROM photos) / (SELECT Count(*) FROM users) AS avg;

+----+
| avg |
+-----+
| 2.5700 |
```

6. What are the top 5 most commonly used hashtags?

```
SELECT
    tags.tag_name,
    COUNT(*) AS total_tags
FROM tags
JOIN photo_tags
    ON tags.id = tag_id
GROUP BY tags.tag_name
ORDER BY total_tags DESC
LIMIT 5
;
```

7. Find users who have liked every single photo on the site

```
SELECT
    users.id,
    username
FROM users
JOIN likes
    ON users.id = likes.user_id
GROUP BY users.id
HAVING COUNT(*) = (SELECT COUNT(*) FROM photos);
```

```
id | username
     Aniya_Hackett
14 | Jaclyn81
21
     Rocio33
24
     Maxwell.Halvorson
36
     Ollie_Ledner37
     Mckenna17
41
     Duane60
54
     Julien_Schmidt
     Mike.Auer39
66
     Nia_Haag
71
75
   | Leslie67
     Janelle.Nikolaus81
76
91 | Bethany20
```

8. Find users who have never commented on a photo

```
SELECT
    users.id,
    username,
    CASE
        WHEN ISNULL(comments.id) THEN 0
        ELSE COUNT(*)
    END AS n_comments
FROM users
LEFT JOIN comments
    ON users.id = comments.user_id
GROUP BY users.id
HAVING n_comments = 0
;
```

```
id | username
                          | n_comments
     Kenton_Kirlin
     Kasandra_Homenick
                                     Θ
     Eveline95
     Tierra.Trantow
25
                                     0
   | Jaime53
                                     Θ
     Pearl7
                                     Θ
     David.Osinski47
                                     0
     Morgan.Kassulke
                                     0
     Mariano_Koch3
                                     Θ
     Linnea59
     Aurelie71
                                     Θ
     Cesar93
     Florence99
                                     Θ
    Franco_Keebler64
                                     Θ
     Hulda.Macejkovic
     Donald.Fritsch
                                     Θ
     Darby_Herzog
81
     Esther.Zulauf61
                                     Θ
     Bartholome.Bernhard
     Delfina_VonRueden68
                                     Θ
88 | Clint27
                                     0
   | Jessyca_West
89
                                     Θ
  | Esmeralda.Mraz57
                                     0
```

9. Find the percentage of our users who have either never commented on a photo or have commented on every photo

```
((
    COUNT(*)
        users.id,
        username,
            WHEN ISNULL(comments.id) THEN 0
            ELSE COUNT(*)
        END AS n_comments
    FROM users
    LEFT JOIN comments
        ON users.id = comments.user_id
    GROUP BY users.id
    HAVING n_comments = 0
    ) AS no_comments
    COUNT(*)
FROM(
       users.id,
        username
    FROM users
    JOIN comments
        ON users.id = comments.user_id
    GROUP BY users.id
   HAVING COUNT(*) = (SELECT COUNT(*) FROM photos)
) AS all_commented
))
/ COUNT(*) * 100
AS percentage
users
```