Section 15. Working with Lots of Instagram Data

1. Find the 5 oldest users. Reward users who have been around the longest.

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

2. Most Polular Registration Date

```
SELECT
    DAYNAME(created_at) AS day,
    COUNT(*) AS total
FROM users
GROUP BY day
ORDER BY total DESC
LIMIT 1;
```

3. Identify Inactive Users (users with no photos)

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

4. Identify most popular photo (and user who created it)

```
SELECT
   username,
   photos.id,
   photos.image_url,
   COUNT(*) 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;
```

5. How many times does the average user post? Calculate avg number of photos per user

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

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

```
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;
```

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

```
# 正确写法: WHERE 在 GROUP BY 之前（what the selected data you want to group）
SELECT
   username,
   COUNT(*) AS num_likes
FROM users
INNER JOIN likes
   ON users.id = likes.user_id
WHERE number_likes = (SELECT COUNT(*) FROM photos)
GROUP BY likes.user_id;

# 正确写法：HAVING
SELECT
   username,
   COUNT(*) AS num_likes
FROM users
INNER JOIN likes
   ON users.id = likes.user_id
GROUP BY likes.user_id
HAVING number_likes = (SELECT COUNT(*) FROM photos);
```

# The WHERE clause is applied first to the individual rows in the tables or table-valued objects in the Diagram pane. Only the rows that meet the conditions in the WHERE clause are grouped.

# The HAVING clause is then applied to the rows in the result set. Only the groups that meet the HAVING conditions appear in the query output. You can apply a HAVING clause only to columns that also appear in the GROUP BY clause or in an aggregate function.