Section slides: <http://webdev.slides.com/coltsteele/mysql-106#/19>

* In this section we’ll be inserting a ton of data into our Instagram database schema, then asking some questions that we’ll need to answer using the techniques we’ve learned in this class

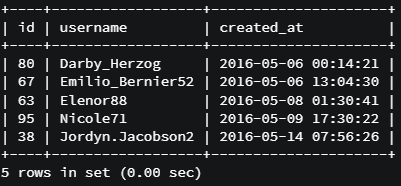
# Inserting the Dataset

* For this section we’ll use the ig\_clone\_data.sql file that the instructor has created
* It consists of thousands of likes, 21 different hashtags, a couple hundred photos, and 100 users

# Instagram Clone Challenge 1 – The 5 Most-Tenured Users

* Suppose we want to reward our users who have been around the longest. **Find the 5 oldest users**.
* This can be accomplished simply be listing out all the users, sorting them by the **created\_at** timestamp, and limiting the output to 5





* Solution code

SELECT \*

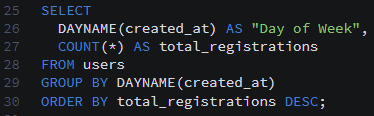
FROM users

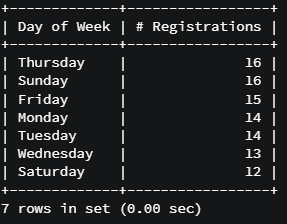
ORDER BY created\_at

LIMIT 5;

# Instagram Clone Challenge 2 – The most popular day of the week to register on

* For this exercise, we want to schedule an ad campaign that will hit the greatest number of new users. Therefore, we want to determine the most popular day of the week that people register on, so we know what day of the week to run on ad on.
* We can determine this by looking at the *users* table and using string functions to determine what day of the week they registered on. The relevant function is DAYNAME(), which returns the day of the week given a DATE, DATETIME, or TIMESTAMP. We’ll also use GROUP BY to group by the name of the day, and the COUNT() function to count how many times that name of the day occurred (and thus the number of registrations on that day).
* We will then ORDER BY that count to determine which day is most popular





* Thus, it appears that Thursday and Sunday are tied for the most popular day of the week to register on.
* Instructor solution code:

SELECT

DAYNAME(created\_at) AS day,

COUNT(\*) AS total

FROM users

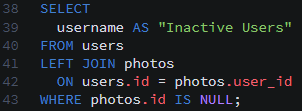
GROUP BY day

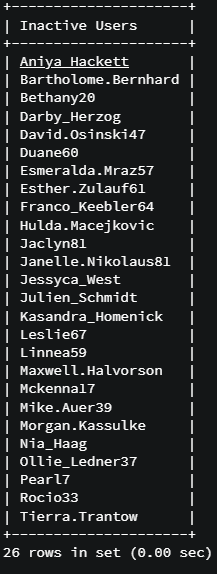
ORDER BY total DESC

LIMIT 2;

# Instagram Clone Challenge 3: Users who have never posted a photo

* We want to send an email campaign out to our inactive users who have never posted a photo
  + Note that a user can be active on Instagram without ever posting photos. They can be “lookers” instead
* We’ll need to work with the *users* and *photos* table here. We’ll need to create a LEF JOIN that lists all users, even those that do not have any photos. Those users who have no photos will have NULL values on the join. We can use this property to select only for those users





* Instructor solution

SELECT username

FROM users

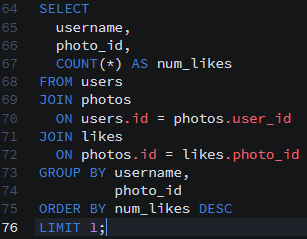
LEFT JOIN photos

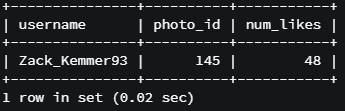
ON users.id = photos.user\_id

WHERE photos.id IS NULL;

# Instagram Clone Challenge 4 – The Most Popular Photo

* In this challenge, we want to find the user who has the most-liked photo. That person wins a prize!
* This will again require a join, this time using the *users*, *photos*, and *likes* tables. In the approach here, we join all three tables and then find the user-photo\_id combo that appears most often.





* Another approach could have been to find the photo with the most likes first (just group the *likes* table by photo\_id), then use that photo\_id and the *photos* table to find the corresponding user
* Instructor solution

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;