**Warbler Log**

**Step One:**

1. *The four tables in the Warbler Database – Follow(“follows”), Like(“likes”), User(“users”), Message(“messages”) – with their rows and respective attributes. Relationships are annotated and indicated by the arrows (dashed arrows indicate “many” from that table, i.e. “many” Messages to one User.*



1. *Note that the follows table has an unusual arrangement: it has two foreign keys to the same table. Why?*

The follows table has two foreign keys to facilitate and separate the ID for a user being followed and a user that is following. The most crucial piece is that both FK’s are also primary keys. This forces both to be unique from each other, but also equally important in terms of querying the follows table for its data. Without both FK’s and without both being PK’s, the table would simply be a pile of connections between users that either follow another user or are being followed, but without an indicative distinction between the two.

### **Step Seven: Research and Understand Login Strategy**

Look over the code in [***app.py***](http://app.py) related to authentication.

* How is the logged in user being kept track of?
  + The ‘’’g.user’’’ variable and session[CURR\_USER\_KEY]
* What is Flask’s ***g*** object?
  + It is an object for storing data within the context of the Flask app. It is comparable to Javascripts “this” or Python’s “self” objects.
* What is the purpose of ***add\_user\_to\_g ?***
  + This function serves the purpose of storing the current, logged-in user in the ‘g object’ for quick and consistent access throughout the application context.
* What does ***@app.before\_request*** mean?
  + This is a built-in, Flask route decorator which Flask executes immediately before every route request. For example, in the Warbler app, ***@app.before\_request*** is used to verify the login status of the session. Once verified, it can accurately perform the operations of the requested route.