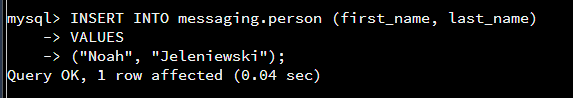
SQL Milestone Document

Noah Jeleniewski

Southern New Hampshire University

The database being worked with in this artifact refinement is called the messaging database. To prove my competence with SQL concepts, I will engage in several common SQL tasks, starting pretty simple and ending with more abstract and complicated SQL concepts. Each task will clearly indicate what is supposed to happen and will have visual evidence of it occurring. Let’s begin:

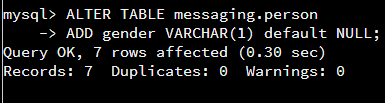
1. Insert a Record into the person table (within the messaging database):



After this, we can see that my name has been added into the table:



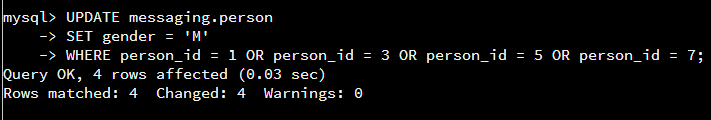
2. Alter the person table (add a custom column):

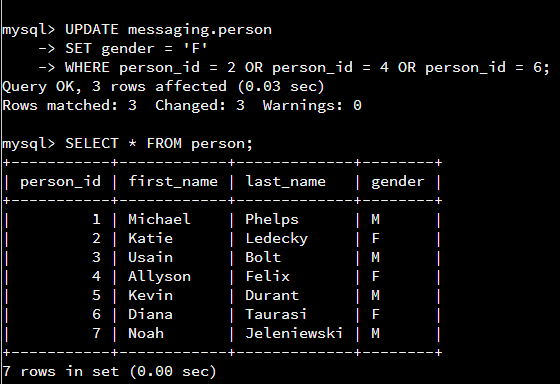


As we can see, I added a gender column, which is one character to indicate M, F, or O (other). It defaults to NULL because it isn’t a required field. Here’s what the person table looks like afterwards:



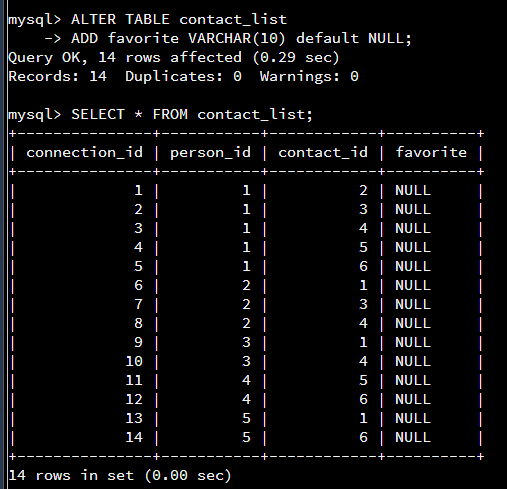
3. Update the gender column for each person (I did this by using two update statements):





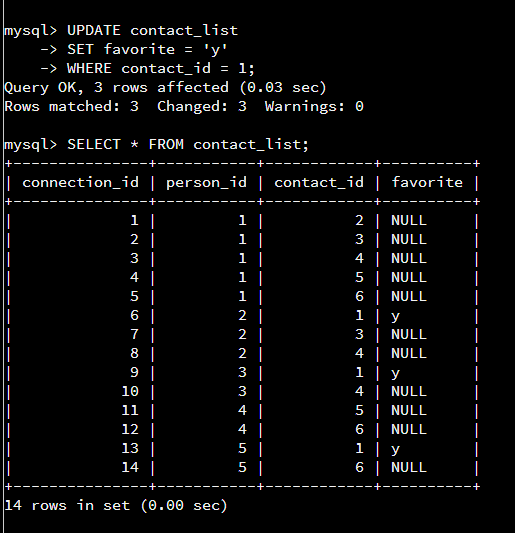
As we can see, every person has had their gender updated accurately.

4. Created a ‘favorite’ field into the contact\_list table:

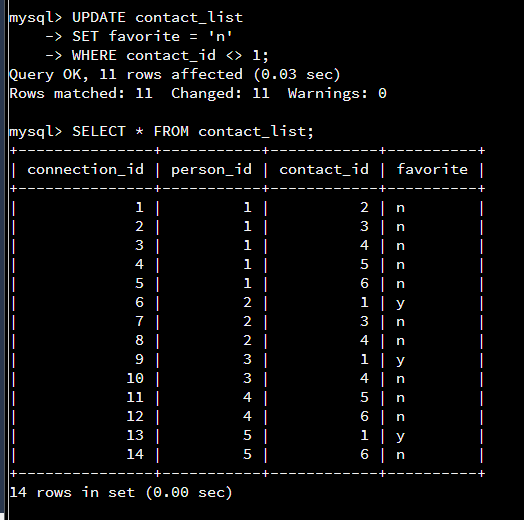


This table will be used for select queries later. The favorite column defaults to NULL since it isn’t a required field, and is a VARCHAR(10).

5. Update the favorite column to ‘y’ where contact\_id = 1:

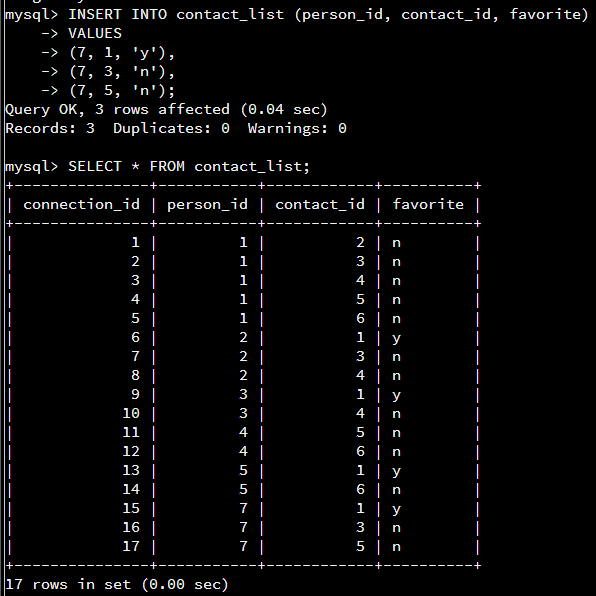


6. Update the favorite column to ‘n’ where contact\_id does NOT = 1:



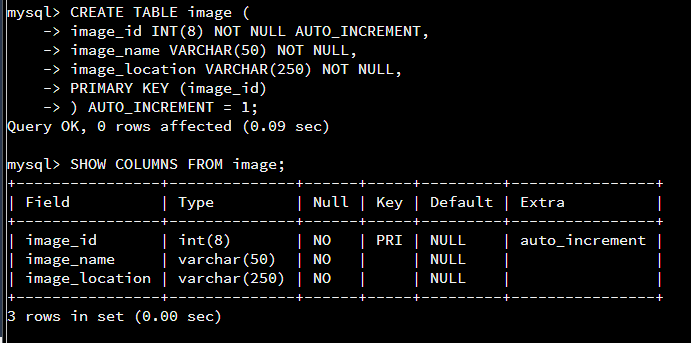
Now the table is fully updated.

7. Add yourself (me) to the contact\_list table with 3 contact id’s (note: I am person\_id 7):



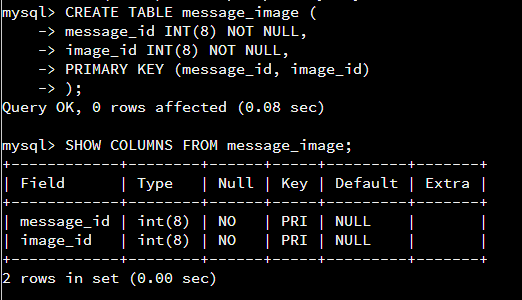
The first contact\_id is 1 (Michael Phelps), which is why the favorite column = ‘y’ for that row. The other two contacts have ‘n’ in that column.

8. Create a new table called image, with an image\_id field, image\_name field, and image\_location field:

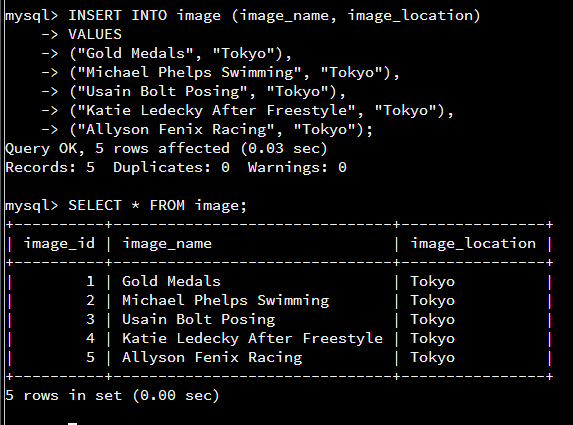


image\_name and image\_location both default to NOT NULL, meaning they are required fields.

9. Create a table called message\_image table (this is a simple intersection table between tables message and image on message\_id and image\_id):



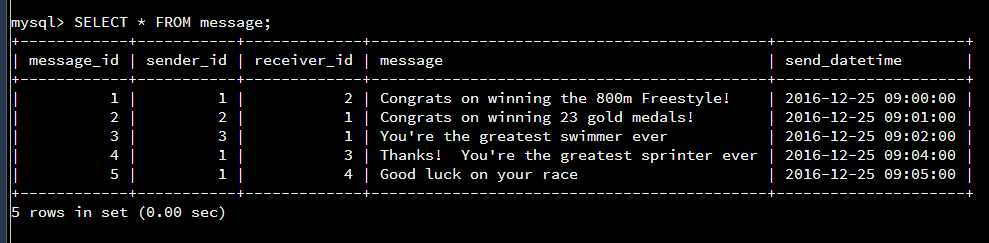
10. Populate the image table with example names/locations:



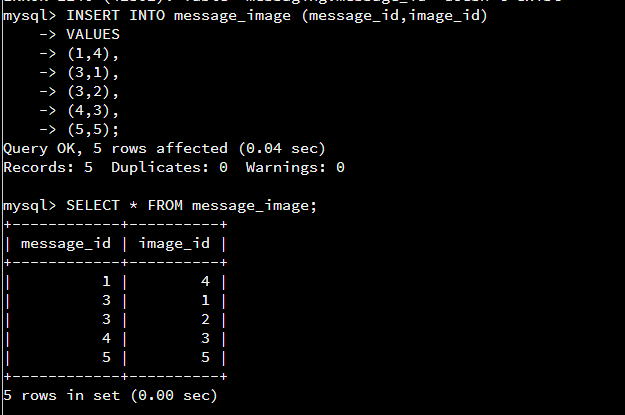
These image names and locations will be referenced later.

11. Insert 5 new records into the message\_image table, based on information from the message table:

Information:

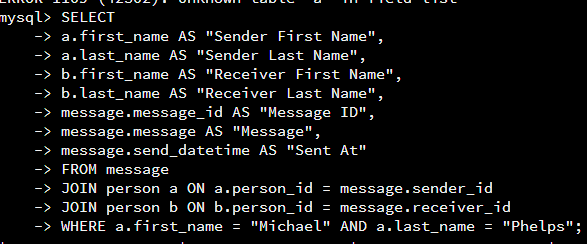


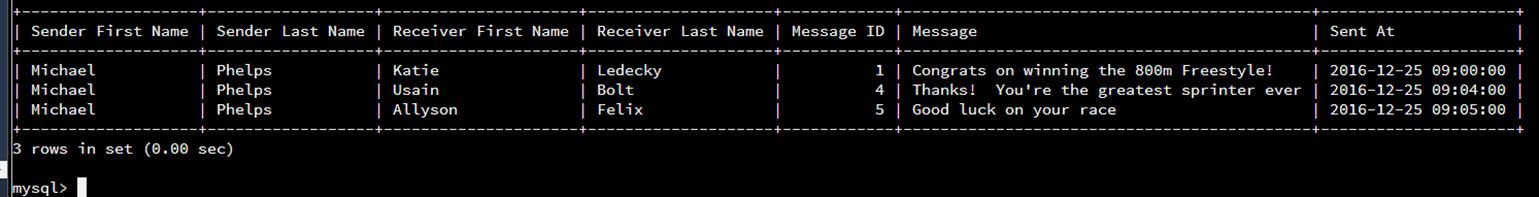
The insert statements:



These id’s line up the messages with the image names in the image table. To associate a message with an image it needed to make logical sense. I checked the message table to see which message maps with which image\_id, and then updated the message\_image intersection table based on this knowledge and the image\_ids from the image table to create messages with images that make sense when looking at it.

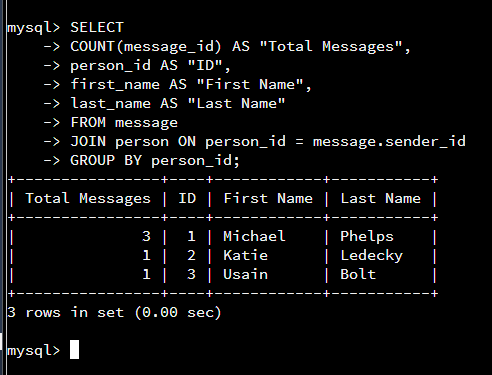
12. Find all of the messages that Michael Phelps has sent. This is where it starts getting complicated, as we reference the same variables within the same select query. Here’s the result:



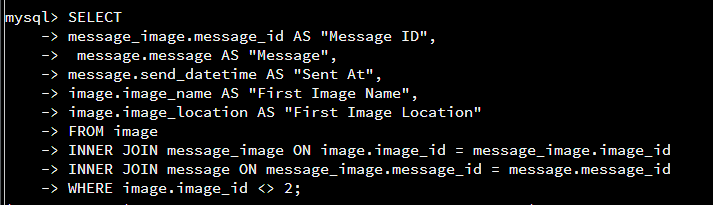
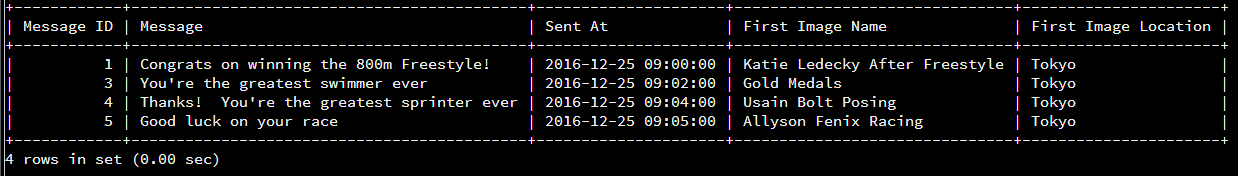


As we can see, this complex select query does display the three messages that Phelps has sent. The first one is to Katie Ledecky, who he congratulates on winning the 800m Freestyle. Then, one to Usain Bolt, who he calls the best sprinter ever. Finally, he texts Allyson Felix, who he wishes luck on her race.

13. Find the number of messages sent for every person (this makes great use of the COUNT() function):

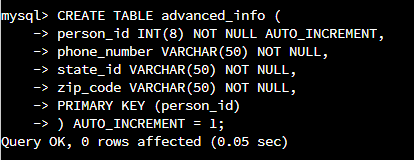


14. Find all of the messages that have AT LEAST one image associated with them using INNER JOINS. This makes use of three different tables (message, message\_image, image) to make it all work. The idea for this is to find all messages with images associated with them, BUT for any messages with TWO images, to only show the FIRST image. Here’s the SQL statement and result:



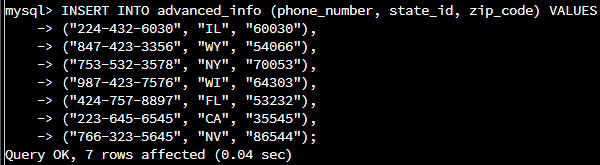
As we can see, each message has been correctly associated with the name of the image associated with it, and its location as well (which in this case is always set to be Tokyo). We can easily check this through common sense (for example, message id 4 is “Thanks! You’re the greatest sprinter ever”, and the image name is “Usain Bolt Posing”).

14. Create a table called advanced\_info, which includes a phone number, state ID, and zip code:

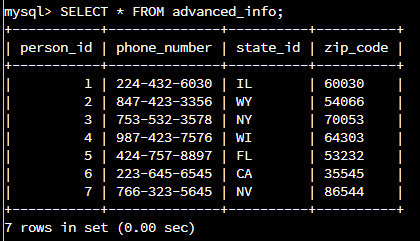


This table will be populated and referenced in the next tasks.

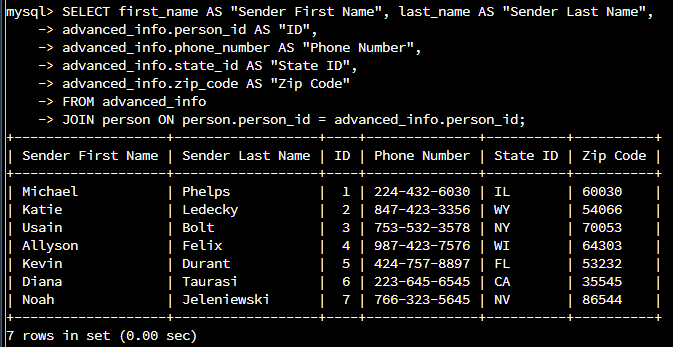
15. Populate the advanced\_info table with relevant fields for each person in the person table:



Here’s the select result:



16. Display the phone number, state ID, and zip code for every person in the person table:



17. Lastly, display the phone number, state ID, and zip code ONLY for people named “Michael Phelps” OR “Noah Jeleniewski”:

