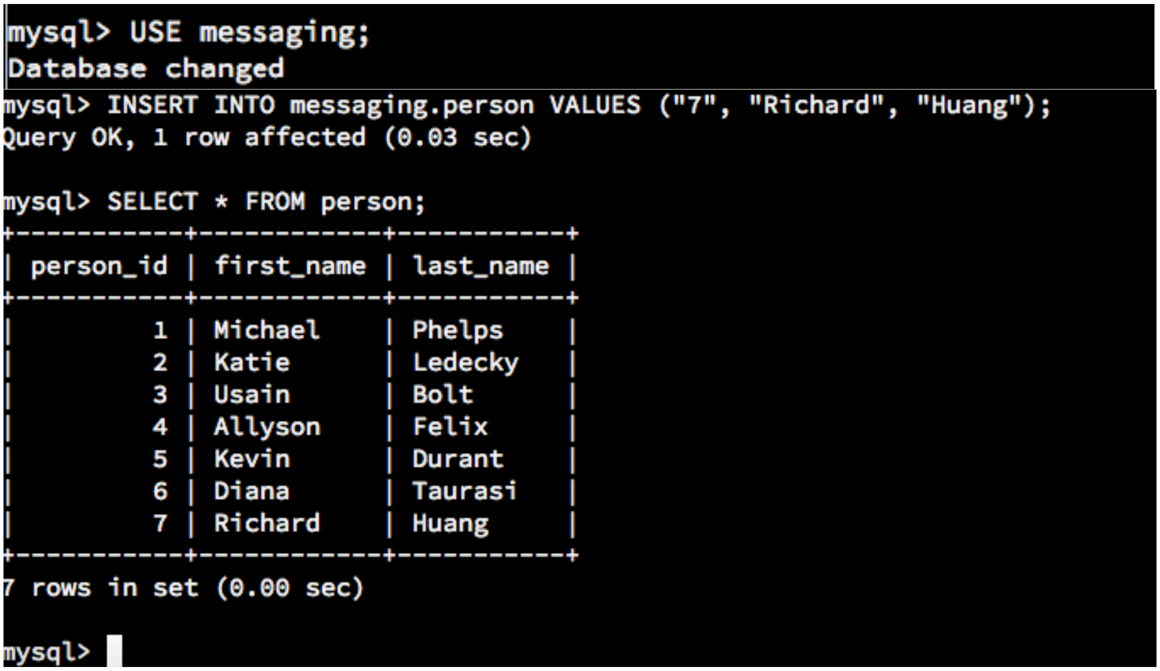
DAD-220 Final Project
Richard Huang
October 14, 2018

Task 1: Insert Record to the Person Table



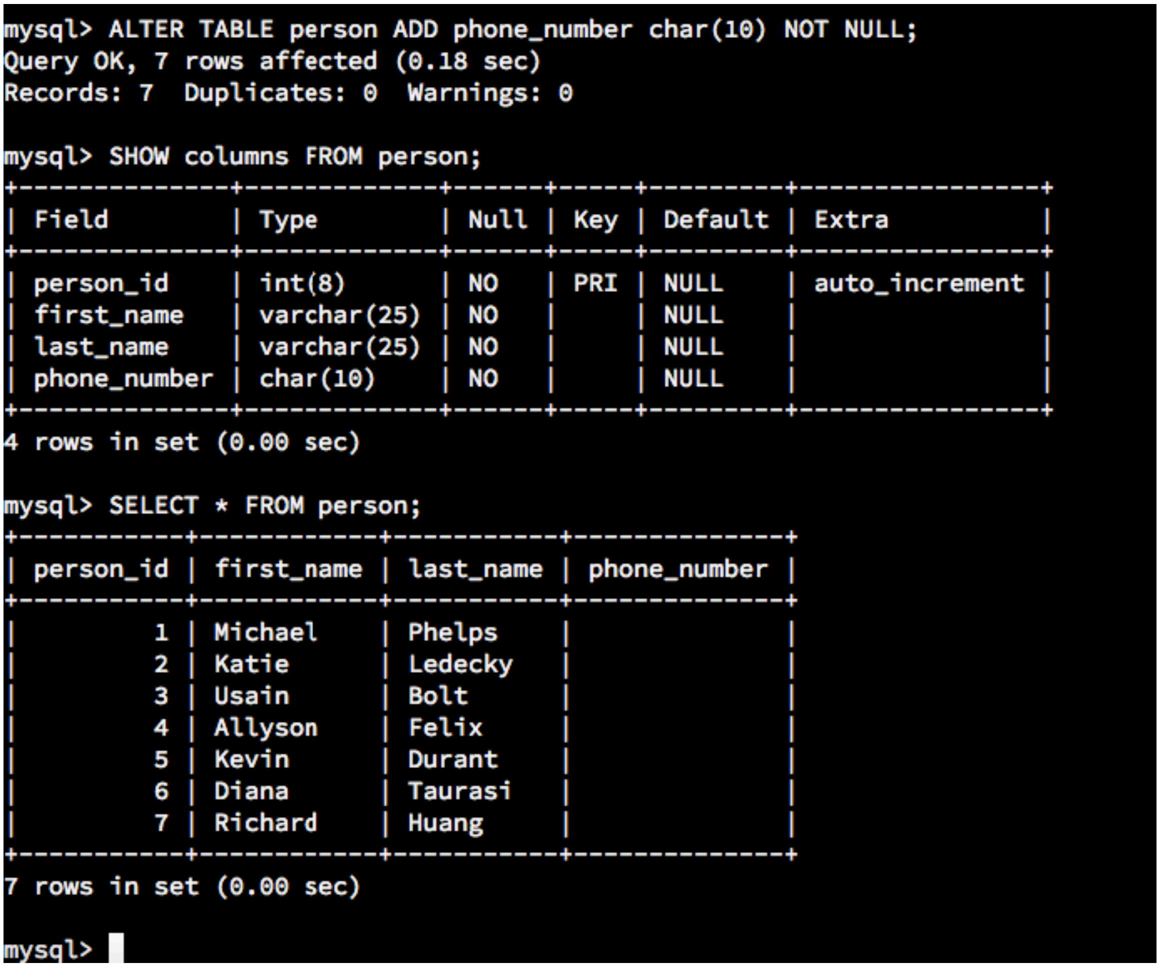
```
mysql> USE messaging;
Database changed
mysql> INSERT INTO messaging.person VALUES ("7", "Richard", "Huang");
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM person;
+-----------+------------+-----------+
| person_id | first_name | last_name |
+-----------+------------+-----------+
|         1 | Michael    | Phelps    |
|         2 | Katie      | Ledecky   |
|         3 | Usain      | Bolt      |
|         4 | Allyson    | Felix     |
|         5 | Kevin      | Durant    |
|         6 | Diana      | Taurasi   |
|         7 | Richard    | Huang     |
+-----------+------------+-----------+
7 rows in set (0.00 sec)

mysql> 
```

Explanation:

Task one required adding my-self as new record in the person table. First, I used the messaging database by entering the “USE messaging;” command, and I used the INSERT INTO statements to complete the task. The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

Task 2: Alter the Person Table



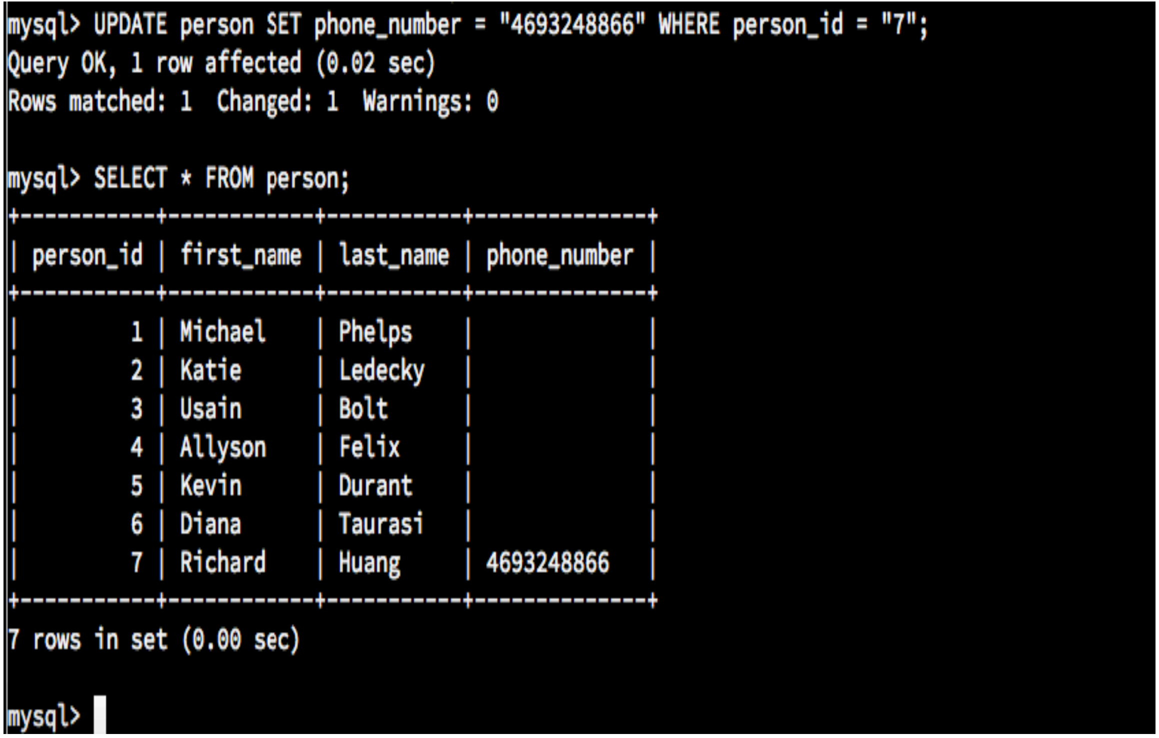
```
mysql> ALTER TABLE person ADD phone_number char(10) NOT NULL;
Query OK, 7 rows affected (0.18 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> SHOW columns FROM person;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| person_id    | int(8)      | NO   | PRI | NULL    | auto_increment |
| first_name   | varchar(25) | NO   |     | NULL    |                |
| last_name    | varchar(25) | NO   |     | NULL    |                |
| phone_number | char(10)    | NO   |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM person;
+-----------+------------+-----------+--------------+
| person_id | first_name | last_name | phone_number |
+-----------+------------+-----------+--------------+
|         1 | Michael    | Phelps    |              |
|         2 | Katie      | Ledecky   |              |
|         3 | Usain      | Bolt      |              |
|         4 | Allyson    | Felix     |              |
|         5 | Kevin      | Durant    |              |
|         6 | Diana      | Taurasi   |              |
|         7 | Richard    | Huang     |              |
+-----------+------------+-----------+--------------+
7 rows in set (0.00 sec)

mysql> 
```

Explanation:

Task two required altering the “person” table with adding a new column. I used ALTER TABLE statement to add a new column and named it “phone_number” with CHAR (10) type and NOT NULL. The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

Task 3: Update Records in the Person Table



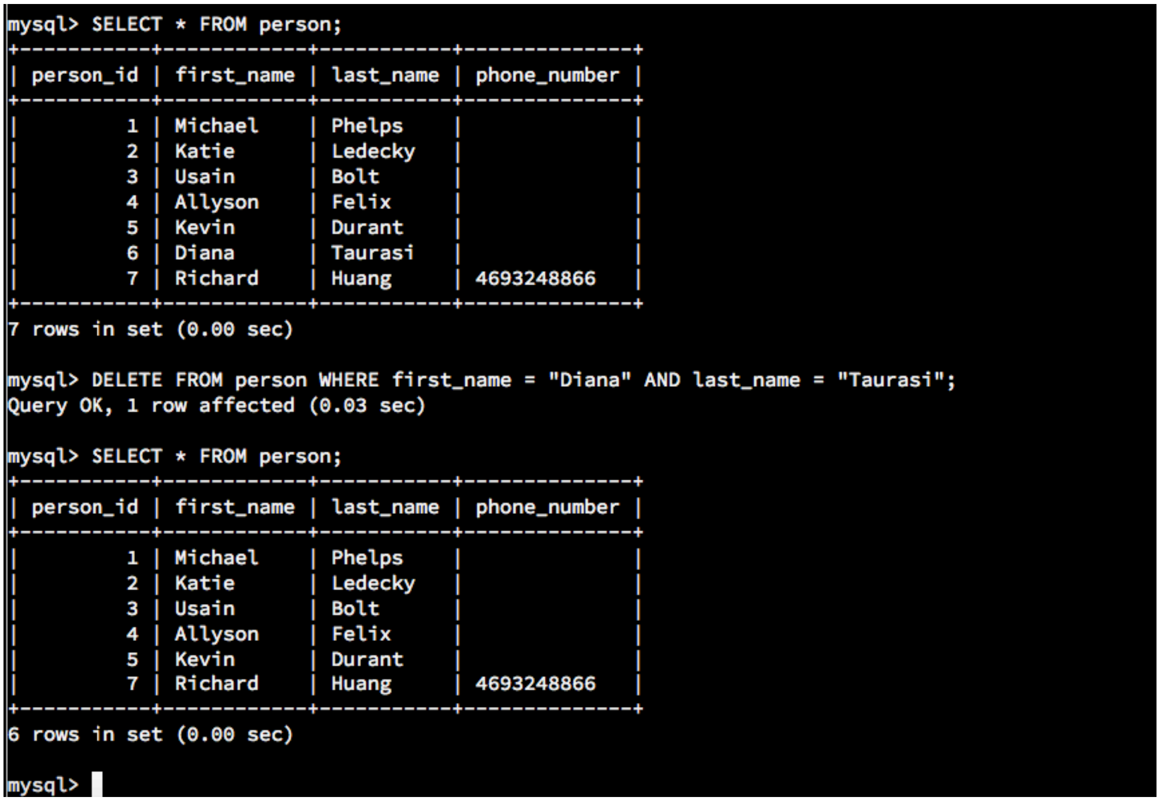
```
mysql> UPDATE person SET phone_number = "4693248866" WHERE person_id = "7";
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM person;
+-----------+------------+-----------+--------------+
| person_id | first_name | last_name | phone_number |
+-----------+------------+-----------+--------------+
|         1 | Michael    | Phelps    |              |
|         2 | Katie      | Ledecky   |              |
|         3 | Usain      | Bolt      |              |
|         4 | Allyson    | Felix     |              |
|         5 | Kevin      | Durant    |              |
|         6 | Diana      | Taurasi   |              |
|         7 | Richard    | Huang     | 4693248866   |
+-----------+------------+-----------+--------------+
7 rows in set (0.00 sec)

mysql> 
```

Explanation:

Task three required updating the record on the person table, and I entered UPDATE statement to update the "phone_number" record to add my phone number to the person table. The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

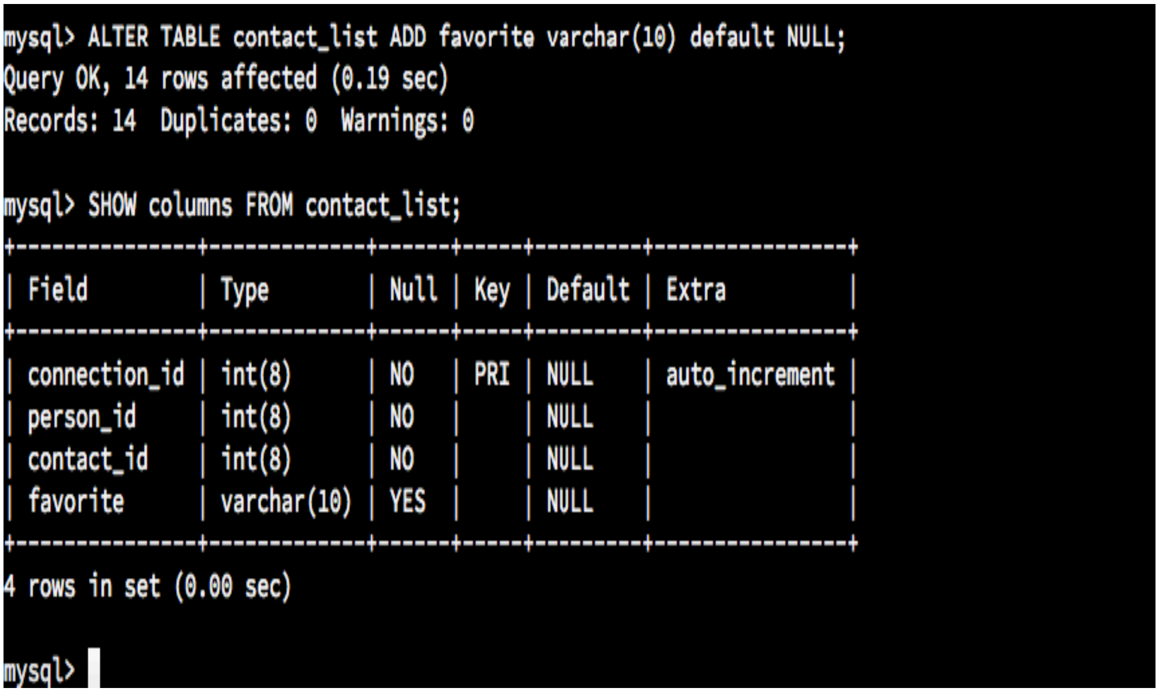
Task 4: Delete Records from Person Table



```
mysql> SELECT * FROM person;
+-----------+------------+-----------+--------------+
| person_id | first_name | last_name | phone_number |
+-----------+------------+-----------+--------------+
|         1 | Michael    | Phelps    |              |
|         2 | Katie      | Ledecky   |              |
|         3 | Usain      | Bolt      |              |
|         4 | Allyson    | Felix     |              |
|         5 | Kevin      | Durant    |              |
|         6 | Diana      | Taurasi   |              |
|         7 | Richard    | Huang     | 4693248866   |
+-----------+------------+-----------+--------------+
7 rows in set (0.00 sec)

mysql> DELETE FROM person WHERE first_name = "Diana" AND last_name = "Taurasi";
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM person;
+-----------+------------+-----------+--------------+
| person_id | first_name | last_name | phone_number |
+-----------+------------+-----------+--------------+
|         1 | Michael    | Phelps    |              |
|         2 | Katie      | Ledecky   |              |
|         3 | Usain      | Bolt      |              |
|         4 | Allyson    | Felix     |              |
|         5 | Kevin      | Durant    |              |
|         7 | Richard    | Huang     | 4693248866   |
+-----------+------------+-----------+--------------+
6 rows in set (0.00 sec)

mysql> 
```

Explanation:
Task four required deleting the record name “Diana Taurasi”, and I entered the DELETE statement where first_name = “Diana” and last_name = “Taurasi”. The result came out successfully, and the record with person_id 6 had deleted.

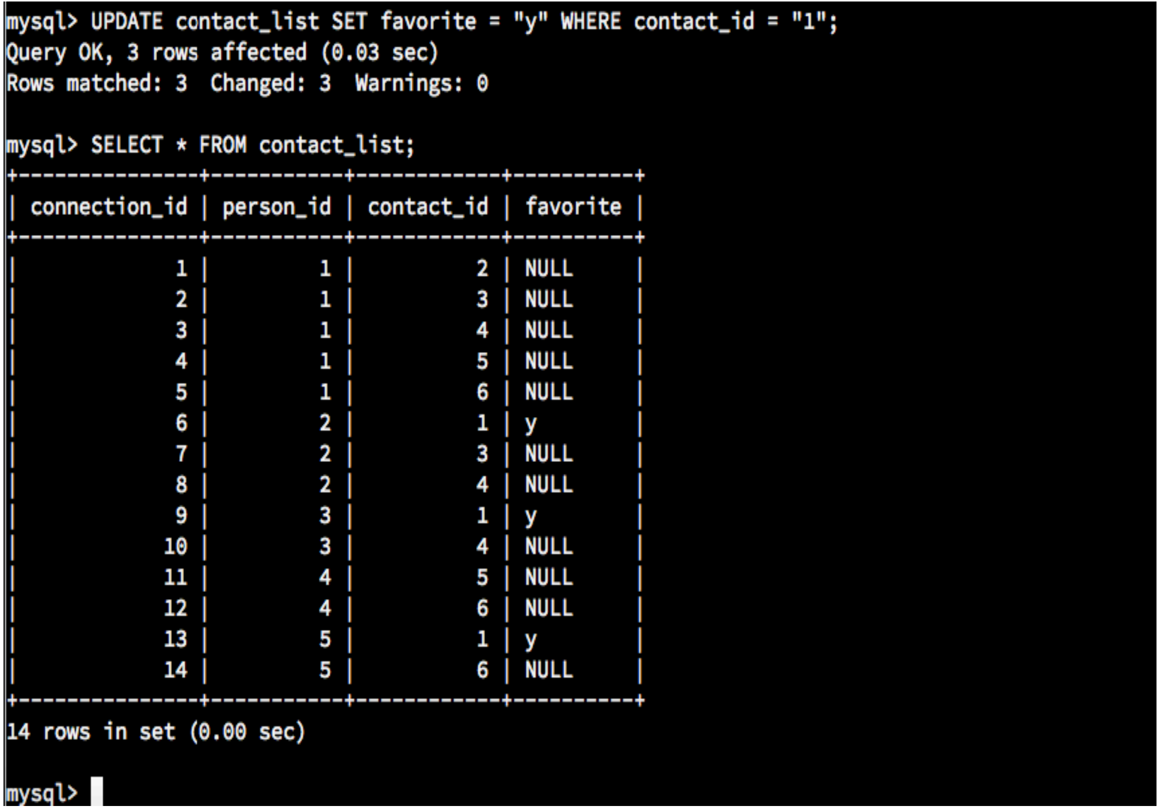
Task 5: Alter the Contact List Table



```
mysql> ALTER TABLE contact_list ADD favorite varchar(10) default NULL;
Query OK, 14 rows affected (0.19 sec)
Records: 14  Duplicates: 0  Warnings: 0

mysql> SHOW columns FROM contact_list;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| connection_id | int(8)      | NO   | PRI | NULL    | auto_increment |
| person_id     | int(8)      | NO   |     | NULL    |                |
| contact_id    | int(8)      | NO   |     | NULL    |                |
| favorite      | varchar(10) | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql> 
```

Explanation:
Task five required altering contact list table to add new column name “favorite”, varchar(10) type and NOT REQUIRED. I entered the ALTER TABLE statement with VARCHAR (10) default NULL. The result came out successfully, and favorite column had added with required type and others.

Task 6: Update Records in the Contact List Table

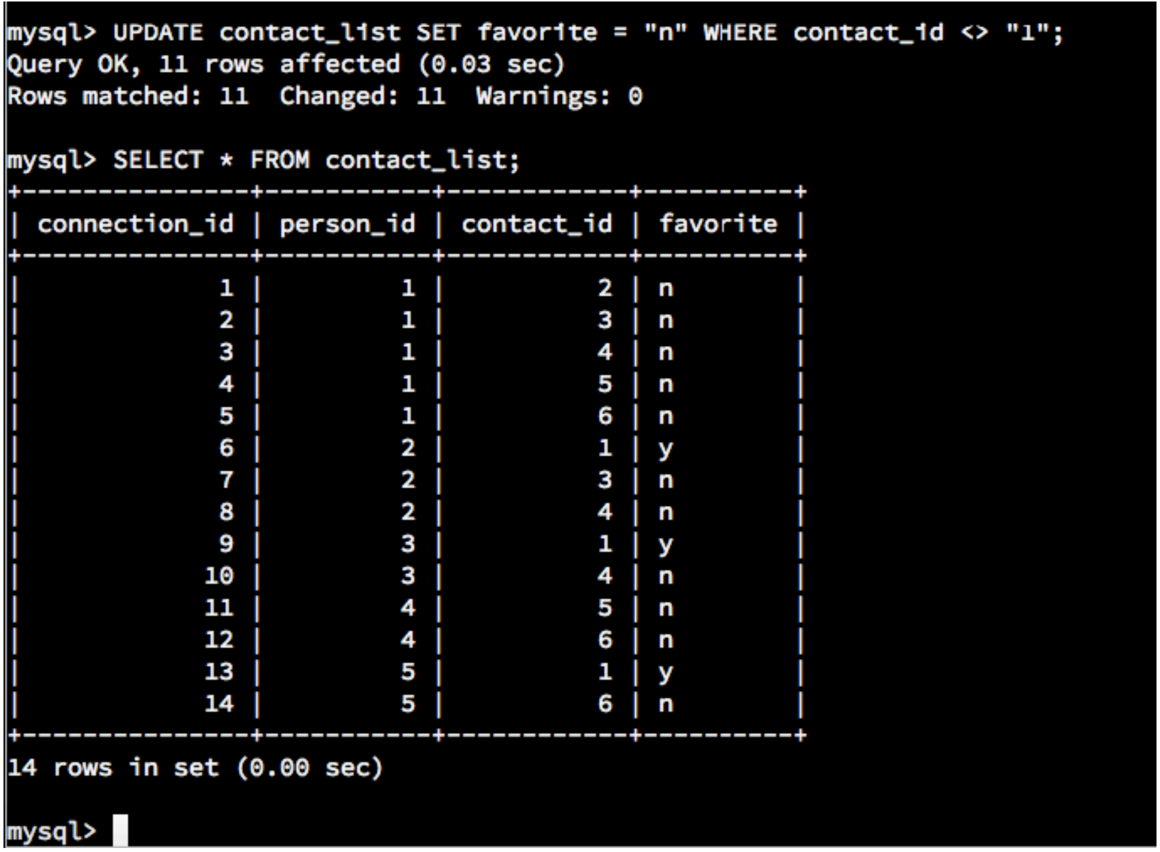


```
mysql> UPDATE contact_list SET favorite = "y" WHERE contact_id = "1";
Query OK, 3 rows affected (0.03 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> SELECT * FROM contact_list;
+---------------+-----------+------------+----------+
| connection_id | person_id | contact_id | favorite |
+---------------+-----------+------------+----------+
|             1 |         1 |          2 | NULL     |
|             2 |         1 |          3 | NULL     |
|             3 |         1 |          4 | NULL     |
|             4 |         1 |          5 | NULL     |
|             5 |         1 |          6 | NULL     |
|             6 |         2 |          1 | y        |
|             7 |         2 |          3 | NULL     |
|             8 |         2 |          4 | NULL     |
|             9 |         3 |          1 | y        |
|            10 |         3 |          4 | NULL     |
|            11 |         4 |          5 | NULL     |
|            12 |         4 |          6 | NULL     |
|            13 |         5 |          1 | y        |
|            14 |         5 |          6 | NULL     |
+---------------+-----------+------------+----------+
14 rows in set (0.00 sec)

mysql> 
```

Explanation:
Task six required updating the new “favorite” column on contact list table, and we needed to set the Mr. Michael Phelps (contact_id =1) as favorite contact. I entered the UPDATE statement and set all contact id 1’s favorite column with “y” (yes). The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

Task 7: Update Records in the Contact List Table

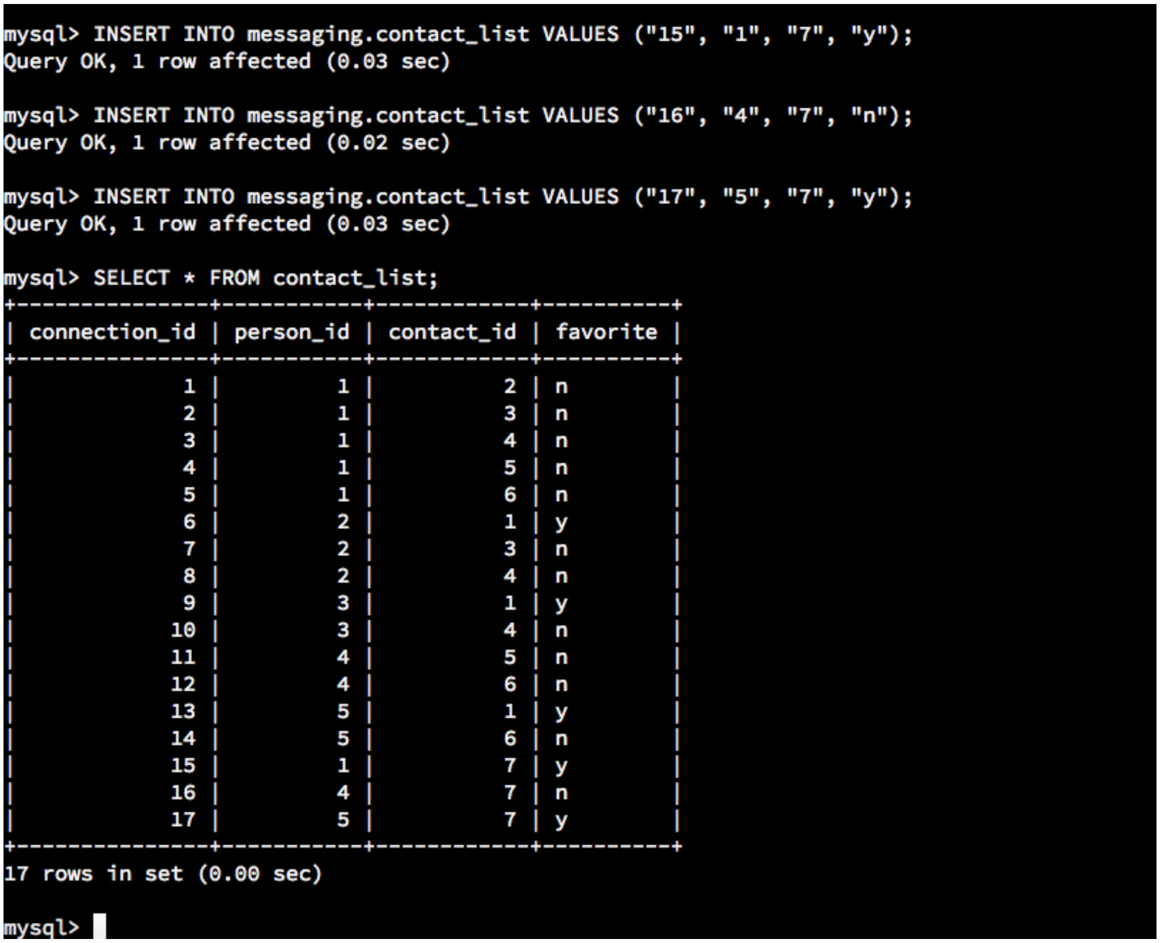


```
mysql> UPDATE contact_list SET favorite = "n" WHERE contact_id <> "1";
Query OK, 11 rows affected (0.03 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> SELECT * FROM contact_list;
+---------------+-----------+------------+----------+
| connection_id | person_id | contact_id | favorite |
+---------------+-----------+------------+----------+
|             1 |         1 |          2 | n        |
|             2 |         1 |          3 | n        |
|             3 |         1 |          4 | n        |
|             4 |         1 |          5 | n        |
|             5 |         1 |          6 | n        |
|             6 |         2 |          1 | y        |
|             7 |         2 |          3 | n        |
|             8 |         2 |          4 | n        |
|             9 |         3 |          1 | y        |
|            10 |         3 |          4 | n        |
|            11 |         4 |          5 | n        |
|            12 |         4 |          6 | n        |
|            13 |         5 |          1 | y        |
|            14 |         5 |          6 | n        |
+---------------+-----------+------------+----------+
14 rows in set (0.00 sec)

mysql>
```

Explanation:
Task seven required us to update remaining records under favorite column, and set other people beside Mr. Michael Phelps (contact_id =1) to not favorite. I entered the UPDATE statement and set “n” (no) to remaining contact ids. The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

Task 8: Insert Records to Contact List Table

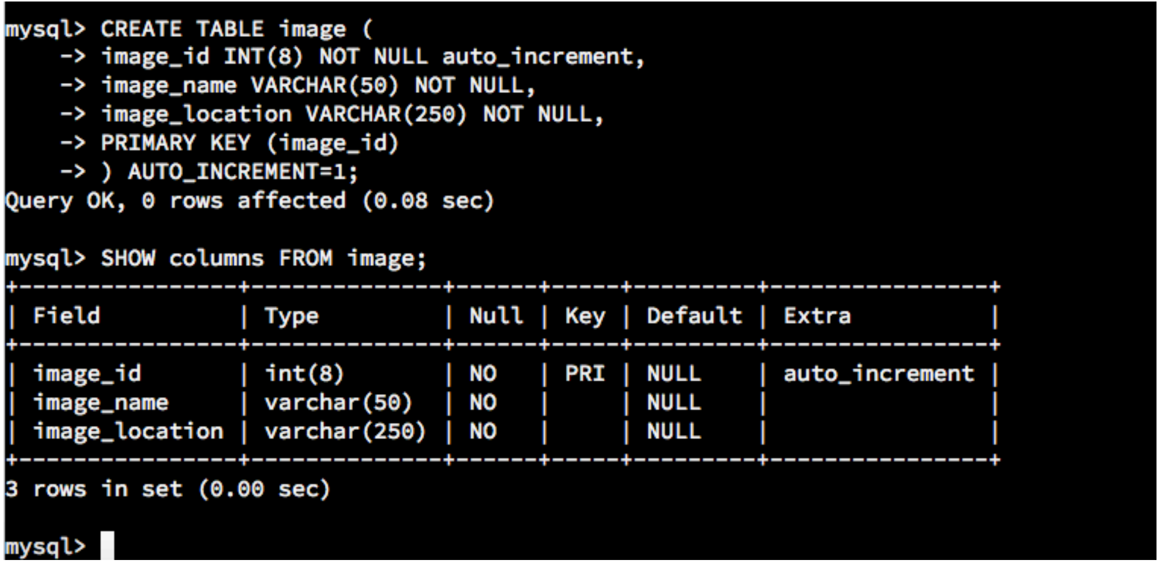


```
mysql> INSERT INTO messaging.contact_list VALUES ("15", "1", "7", "y");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO messaging.contact_list VALUES ("16", "4", "7", "n");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO messaging.contact_list VALUES ("17", "5", "7", "y");
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM contact_list;
+---------------+-----------+------------+----------+
| connection_id | person_id | contact_id | favorite |
+---------------+-----------+------------+----------+
|             1 |         1 |          2 | n        |
|             2 |         1 |          3 | n        |
|             3 |         1 |          4 | n        |
|             4 |         1 |          5 | n        |
|             5 |         1 |          6 | n        |
|             6 |         2 |          1 | y        |
|             7 |         2 |          3 | n        |
|             8 |         2 |          4 | n        |
|             9 |         3 |          1 | y        |
|            10 |         3 |          4 | n        |
|            11 |         4 |          5 | n        |
|            12 |         4 |          6 | n        |
|            13 |         5 |          1 | y        |
|            14 |         5 |          6 | n        |
|            15 |         1 |          7 | y        |
|            16 |         4 |          7 | n        |
|            17 |         5 |          7 | y        |
+---------------+-----------+------------+----------+
17 rows in set (0.00 sec)

mysql> 
```

Explanation:
Task eight required us to insert three new records, and add my-self as a new contact in the contact_list table. I entered INSERT statements with connection id 15, 16, and 17, person id 1, 4 and 5. I had provided the values for “favorite” column under my contact id.

DAD-220 Final Project
Richard Huang
October 14, 2018

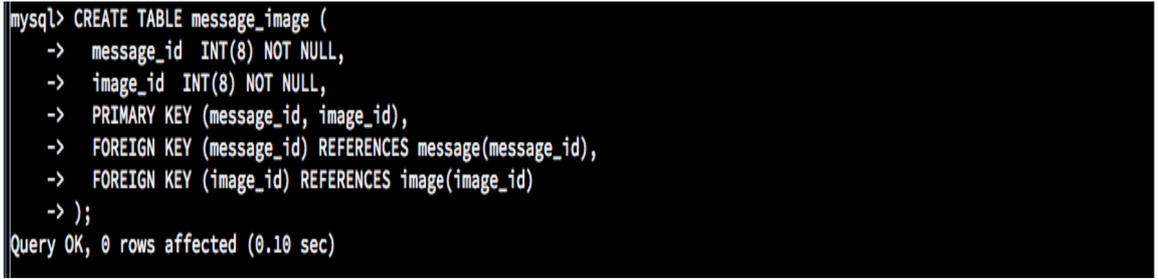
Task 9: Create the Image Table



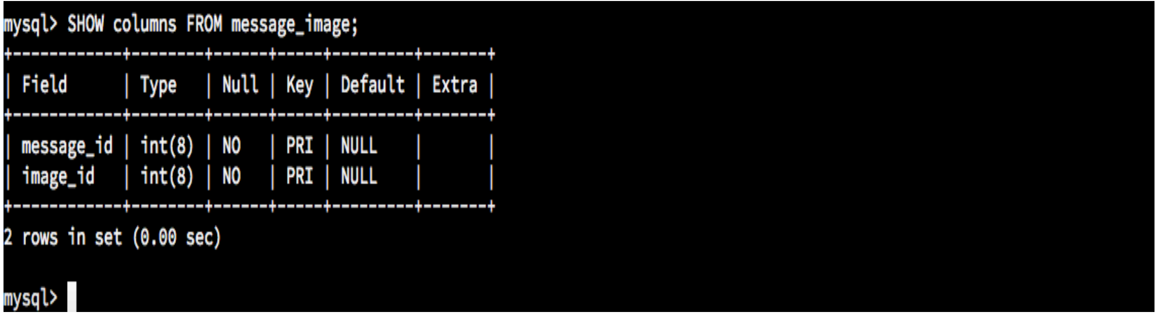
```
mysql> CREATE TABLE image (
    -> image_id INT(8) NOT NULL auto_increment,
    -> image_name VARCHAR(50) NOT NULL,
    -> image_location VARCHAR(250) NOT NULL,
    -> PRIMARY KEY (image_id)
    -> ) AUTO_INCREMENT=1;
Query OK, 0 rows affected (0.08 sec)

mysql> SHOW columns FROM image;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| image_id       | int(8)       | NO   | PRI | NULL    | auto_increment |
| image_name     | varchar(50)  | NO   |     | NULL    |                |
| image_location | varchar(250) | NO   |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> 
```

Explanation:
Task nine required us to create a new table name “image” with column data type and constraints. I entered the CREATE TABLE statements with all the requirements on task nine, and set image_id as primary key. The result came out successfully.

Task 10: Create the Message-Image Intersection Table



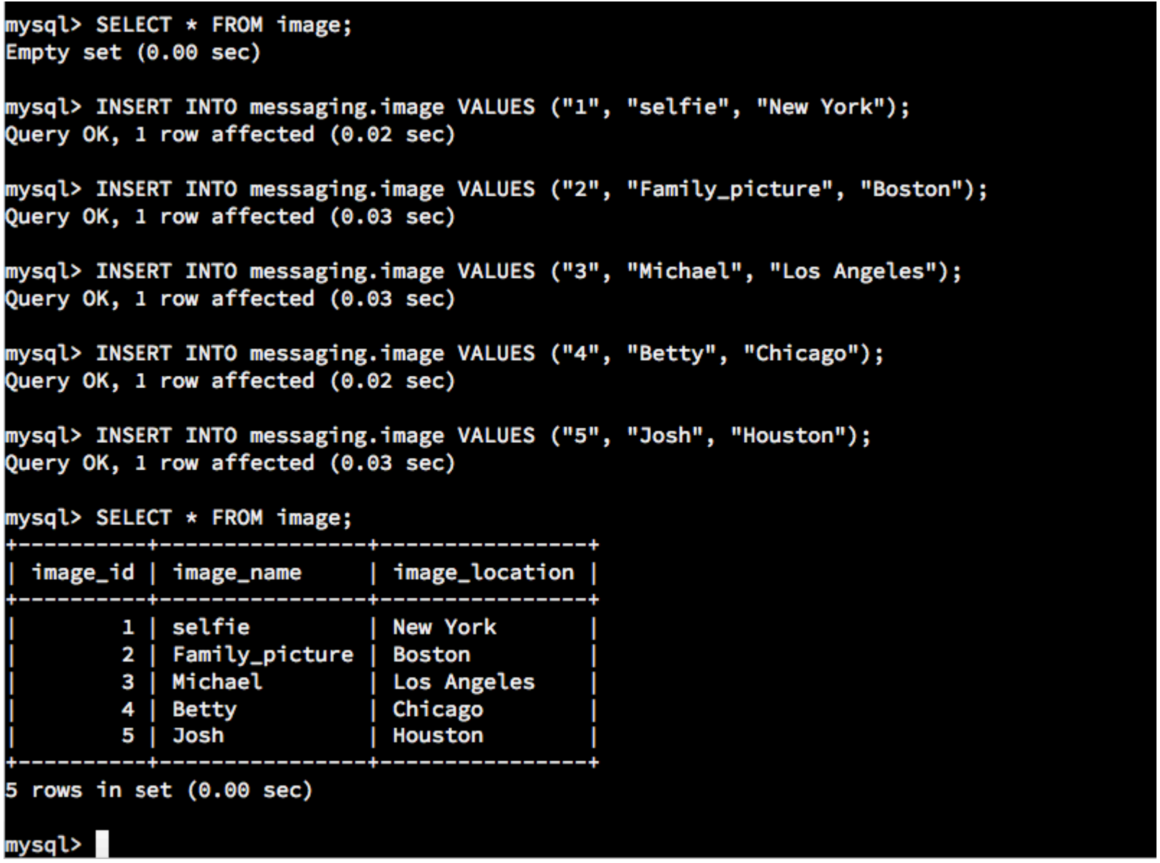
```
mysql> CREATE TABLE message_image (
    ->   message_id  INT(8) NOT NULL,
    ->   image_id  INT(8) NOT NULL,
    ->   PRIMARY KEY (message_id, image_id),
    ->   FOREIGN KEY (message_id) REFERENCES message(message_id),
    ->   FOREIGN KEY (image_id) REFERENCES image(image_id)
    -> );
Query OK, 0 rows affected (0.10 sec)
```



```
mysql> SHOW columns FROM message_image;
+------------+--------+------+-----+---------+-------+
| Field      | Type   | Null | Key | Default | Extra |
+------------+--------+------+-----+---------+-------+
| message_id | int(8) | NO   | PRI | NULL    |       |
| image_id   | int(8) | NO   | PRI | NULL    |       |
+------------+--------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> 
```

Explanation:
Task ten required creating intersection table named “message_image”, and I used CREATE TABLE statements with assigning PRIMARY KEY AND FOREIGN KEY to each column. The result came out successfully, and the table was created with required specifications.

Task 11: Insert Records to Image Table

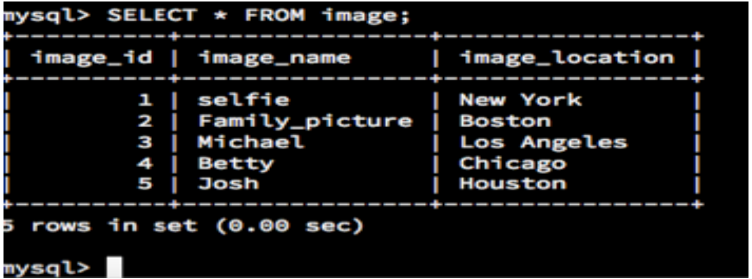


```
mysql> SELECT * FROM image;
Empty set (0.00 sec)

mysql> INSERT INTO messaging.image VALUES ("1", "selfie", "New York");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO messaging.image VALUES ("2", "Family_picture", "Boston");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO messaging.image VALUES ("3", "Michael", "Los Angeles");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO messaging.image VALUES ("4", "Betty", "Chicago");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO messaging.image VALUES ("5", "Josh", "Houston");
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM image;
+----------+----------------+----------------+
| image_id | image_name     | image_location |
+----------+----------------+----------------+
|        1 | selfie         | New York       |
|        2 | Family_picture | Boston         |
|        3 | Michael        | Los Angeles    |
|        4 | Betty          | Chicago        |
|        5 | Josh           | Houston        |
+----------+----------------+----------------+
5 rows in set (0.00 sec)

mysql> 
```

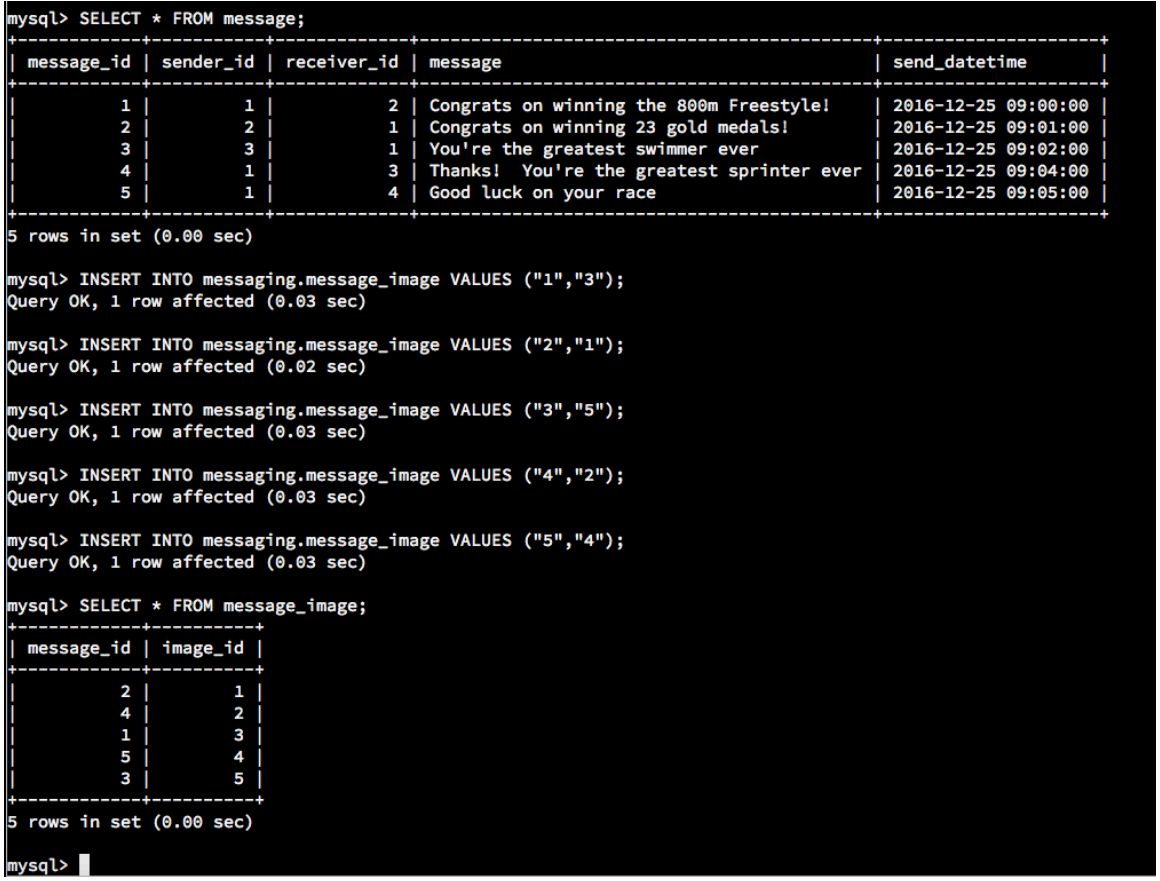
Explanation:
Task eleven required us to insert 5 new records into the "image" table, and I used INERT INTO statements to add required information and values. The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

Task 12: Insert Records to Message-Image Table



```
mysql> SELECT * FROM image;
+----------+---------------+----------------+
| image_id | image_name    | image_location |
+----------+---------------+----------------+
|        1 | selfie        | New York       |
|        2 | Family_picture| Boston         |
|        3 | Michael       | Los Angeles    |
|        4 | Betty         | Chicago        |
|        5 | Josh          | Houston        |
+----------+---------------+----------------+
5 rows in set (0.00 sec)

mysql> 
```



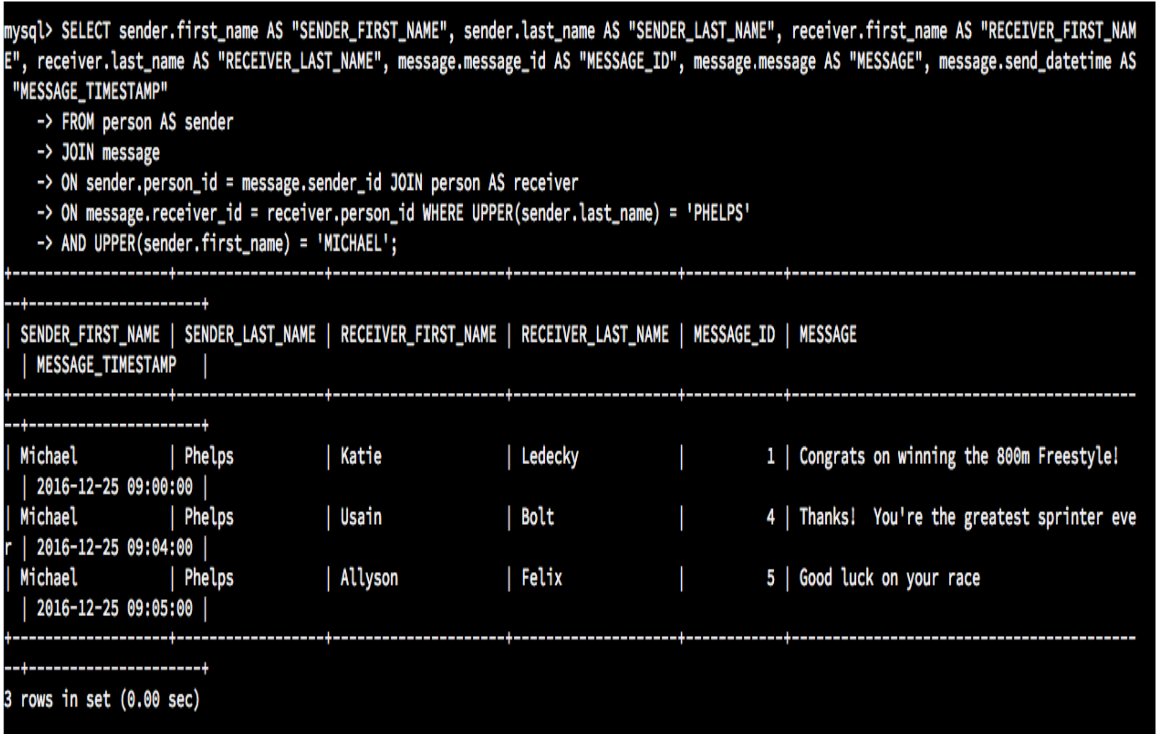
```
mysql> SELECT * FROM message;
+------------+-----------+-------------+---------------------------------------+---------------------+
| message_id | sender_id | receiver_id | message                               | send_datetime       |
+------------+-----------+-------------+---------------------------------------+---------------------+
|          1 |         1 |           2 | Congrats on winning the 800m Freestyle! | 2016-12-25 09:00:00 |
|          2 |         2 |           1 | Congrats on winning 23 gold medals!   | 2016-12-25 09:01:00 |
|          3 |         3 |           1 | You're the greatest swimmer ever      | 2016-12-25 09:02:00 |
|          4 |         1 |           3 | Thanks!  You're the greatest sprinter ever | 2016-12-25 09:04:00 |
|          5 |         1 |           4 | Good luck on your race                | 2016-12-25 09:05:00 |
+------------+-----------+-------------+---------------------------------------+---------------------+
5 rows in set (0.00 sec)

mysql> INSERT INTO messaging.message_image VALUES ("1","3");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO messaging.message_image VALUES ("2","1");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO messaging.message_image VALUES ("3","5");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO messaging.message_image VALUES ("4","2");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO messaging.message_image VALUES ("5","4");
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM message_image;
+------------+----------+
| message_id | image_id |
+------------+----------+
|          2 |        1 |
|          4 |        2 |
|          1 |        3 |
|          5 |        4 |
|          3 |        5 |
+------------+----------+
5 rows in set (0.00 sec)

mysql> 
```

Explanation:

Task twelve required us inserting 5 new records, and added at least one record where at least one of Michael Phelps' messages includes at least one image. I used INSERT INTO statement and assigned all Mr. Michael Phelps messages with images. The result came out successfully.

DAD-220 Final Project
Richard Huang
October 14, 2018

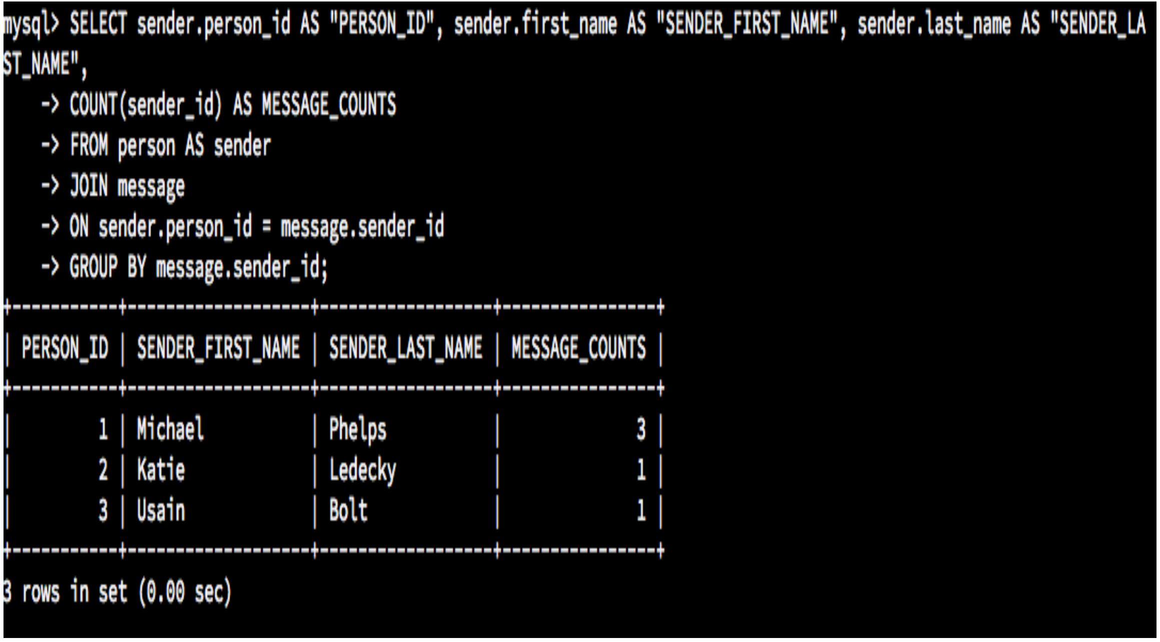
Task 13: Find All of the Messages that Michael Phelps Sent



```
mysql> SELECT sender.first_name AS "SENDER_FIRST_NAME", sender.last_name AS "SENDER_LAST_NAME", receiver.first_name AS "RECEIVER_FIRST_NAM
E", receiver.last_name AS "RECEIVER_LAST_NAME", message.message_id AS "MESSAGE_ID", message.message AS "MESSAGE", message.send_datetime AS
 "MESSAGE_TIMESTAMP"
    -> FROM person AS sender
    -> JOIN message
    -> ON sender.person_id = message.sender_id JOIN person AS receiver
    -> ON message.receiver_id = receiver.person_id WHERE UPPER(sender.last_name) = 'PHELPS'
    -> AND UPPER(sender.first_name) = 'MICHAEL';
+-------------------+------------------+---------------------+--------------------+------------+------------------------------------------
--+---------------------+
| SENDER_FIRST_NAME | SENDER_LAST_NAME | RECEIVER_FIRST_NAME | RECEIVER_LAST_NAME | MESSAGE_ID | MESSAGE
  | MESSAGE_TIMESTAMP   |
+-------------------+------------------+---------------------+--------------------+------------+------------------------------------------
--+---------------------+
| Michael           | Phelps           | Katie               | Ledecky            |          1 | Congrats on winning the 800m Freestyle!
  | 2016-12-25 09:00:00 |
| Michael           | Phelps           | Usain               | Bolt               |          4 | Thanks!  You're the greatest sprinter eve
r | 2016-12-25 09:04:00 |
| Michael           | Phelps           | Allyson             | Felix              |          5 | Good luck on your race
  | 2016-12-25 09:05:00 |
+-------------------+------------------+---------------------+--------------------+------------+------------------------------------------
--+---------------------+
3 rows in set (0.00 sec)
```

Explanation:
Task thirteen required finding all messages that Mr. Michael Phelps sent, and I used the JOIN statements. These commands joined from “person” table with “message” table with displaying the required names and columns. The result came out successfully.

Task 14: Find the Number of Messages Sent for Every Person

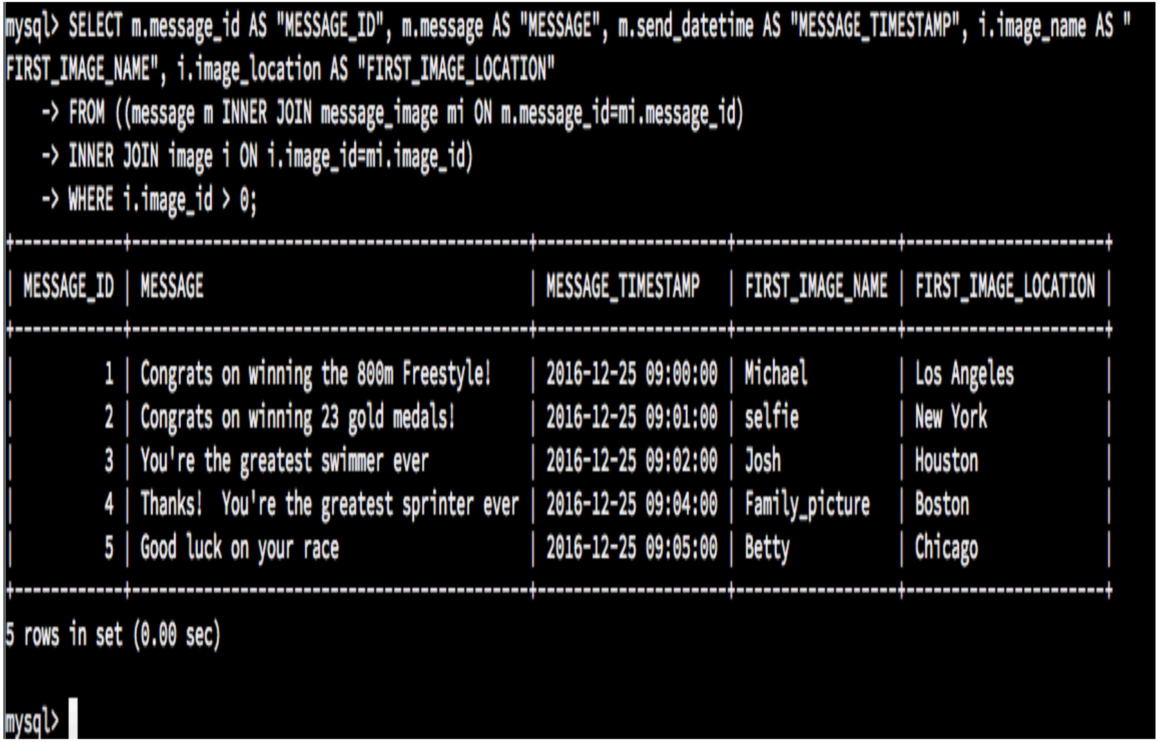


```
mysql> SELECT sender.person_id AS "PERSON_ID", sender.first_name AS "SENDER_FIRST_NAME", sender.last_name AS "SENDER_LA
ST_NAME",
    -> COUNT(sender_id) AS MESSAGE_COUNTS
    -> FROM person AS sender
    -> JOIN message
    -> ON sender.person_id = message.sender_id
    -> GROUP BY message.sender_id;
+-----------+-------------------+------------------+----------------+
| PERSON_ID | SENDER_FIRST_NAME | SENDER_LAST_NAME | MESSAGE_COUNTS |
+-----------+-------------------+------------------+----------------+
|         1 | Michael           | Phelps           |              3 |
|         2 | Katie             | Ledecky          |              1 |
|         3 | Usain             | Bolt             |              1 |
+-----------+-------------------+------------------+----------------+
3 rows in set (0.00 sec)
```

Explanation:

Task fourteen required us to count the message sent, and grouped them by orders. I entered the SELECT, COUNT, JOIN and GROUP BY statements to obtain person id, message counts, and sender’s first and last name. The result came out successfully.

Task 15: Find All of the Messages that Have At Least One Image Attached Using INNER JOINs



```
mysql> SELECT m.message_id AS "MESSAGE_ID", m.message AS "MESSAGE", m.send_datetime AS "MESSAGE_TIMESTAMP", i.image_name AS "
FIRST_IMAGE_NAME", i.image_location AS "FIRST_IMAGE_LOCATION"
    -> FROM ((message m INNER JOIN message_image mi ON m.message_id=mi.message_id)
    -> INNER JOIN image i ON i.image_id=mi.image_id)
    -> WHERE i.image_id > 0;
+------------+--------------------------------------------+---------------------+------------------+----------------------+
| MESSAGE_ID | MESSAGE                                    | MESSAGE_TIMESTAMP   | FIRST_IMAGE_NAME | FIRST_IMAGE_LOCATION |
+------------+--------------------------------------------+---------------------+------------------+----------------------+
|          1 | Congrats on winning the 800m Freestyle!    | 2016-12-25 09:00:00 | Michael          | Los Angeles          |
|          2 | Congrats on winning 23 gold medals!        | 2016-12-25 09:01:00 | selfie           | New York             |
|          3 | You're the greatest swimmer ever           | 2016-12-25 09:02:00 | Josh             | Houston              |
|          4 | Thanks!  You're the greatest sprinter ever | 2016-12-25 09:04:00 | Family_picture   | Boston               |
|          5 | Good luck on your race                     | 2016-12-25 09:05:00 | Betty            | Chicago              |
+------------+--------------------------------------------+---------------------+------------------+----------------------+
5 rows in set (0.00 sec)

mysql>
```

Explanation:

The final task required using INNER JOIN to combine the image table and message table with at least one image attached. I used the INNER JOIN statement to combine all the related tables and showed the required names and columns. The result came out successfully.