

Lab Assignment 3: Working with MySQL

In this assignment we will create some new tables that would help us handle parts of the simple messaging system we discussed earlier. You should download this SQL script from GitHub¹ and open the script file from mysqlworkbench. It contains instructions that would delete any tables we have created on the topic. You will add your own code to this file. When you want to “reset” the database, just execute that script.

We will create three tables in this assignment:

- `profiles` holds information about the users of the system. It contains their username and first and last name.
- `messages` holds information about a message. A row in this table consists of a unique auto-generated message id, the username of the message's sender, the username of the message's recipient, the message's text, and the time and date when the text was sent. It also contains a boolean flag as to whether the message has been read by the recipient, and another field that contains the id of the message that this message responds to, if it is in response to another message.
- `acquaintances` holds information about which user knows which other user.

Make sure to do these assignments in order.

1. The first part of the assignment is to create the three tables.

- Write the `CREATE TABLE` command for a table named `profiles`. It should contain the following fields:
 - `username` should be a variable length character field with at most 20 characters. It should never be null, and it should be unique.
 - `first` should be a variable length character field with at most 40 characters. It is allowed to be null.
 - `last` should be a variable length character field with at most 40 characters. It is not allowed to be null.

Your table should also have the `username` column as its primary key.

- Write a `CREATE TABLE` command for a table named `messages`. It should contain the following fields:
 - `id` should be an integer field that is never null, that is unique, and that is auto-incremented.
 - `sender` should be a variable length character field of length at most 20 characters. It should not be null.
 - `recipient` should be a variable length character field of length at most 20 characters. It should not be null.

¹<https://github.com/skiadas/DataWranglingCourse/blob/gh-pages/assignments/assignment3.sql>

- message should be a variable length character field of length at most 400 characters. It should be not null, and it should have a default value of an empty string.
- is_read should be a boolean field. You can use the BOOLEAN type, or the TINYINT(1) type if BOOLEAN does not work. You can use the values 0 or FALSE for false, and 1 or TRUE for true. The field should be set to be not null, and it should be set to have a default value of FALSE.
- sent_at should be set to have the TIMESTAMP type². It should be set to be not null, and to default to the value CURRENT_TIMESTAMP. Timestamps are recorded as seconds since what is known as the “epoch” (you can find out more than you’ll ever want to know about this here³).
- in_reply_to is an integer field that may be null, and it defaults to NULL.

Your table should have the id as a primary key. It should also have three foreign key constraints:

- One foreign key constraint should specify that the sender field must reference the username field of the profiles table. It should cascade on delete.
 - One foreign key constraint should specify that the recipient field must reference the username field of the profiles table. It should cascade on delete.
 - One foreign key constraint should specify that the in_reply_to field must reference the id field of the messages table. It should be told ON DELETE SET NULL which means that if the referenced message is deleted, then this message should set this entry to NULL rather than also be deleted.
- Write a CREATE TABLE command for a table named acquaintances. The acq table has the following fields:
 - source should be a variable length character field of length at most 20 characters. It should not be null.
 - target should be a variable length character field of length at most 20 characters. It should not be null.

The table should have the pair of source and target as a primary key.

The table also has two foreign key constraints. The two foreign key constraints should specify that the source (respectively target) field references the username field in the profiles array. They should both cascade on delete.

The table’s entries are meant to represent that given a source-target pair, the source user “knows” the target user (and can therefore send them messages).

2. We now need to add some profiles. Write an INSERT command to create three new profiles. One of these profiles should have the username admin, with a null first name and with an Admin last name. Another profile should be yours, and a third should be your favorite Famous Computer Scientist (if you don’t have one, now is a good time to find one). We will abbreviate this scientist in the rest as FCS.

²<https://dev.mysql.com/doc/refman/5.5/en/date-and-time-type-overview.html>

³https://en.wikipedia.org/wiki/Unix_time

3. Write an INSERT command that will add one acquaintance pair to the acquaintances table, namely it should say that you know the famous computer scientist.
4. We want to establish that the admin user knows all other users in the system. Start by constructing a SELECT query that will show you the usernames of all users except for the admin user, then use that query inside an INSERT query that adds entries to the acquaintances table. The entries should contain the admin user as the source, and should have all other users as the target. You **must** use an INSERT-SELECT combination for this, don't use INSERT-VALUES to explicitly list the other two users in your system. Your code should work just as well if there were 100 or 1000 users in the system.
5. Create an INSERT command that adds a message, sent by you to the FCS, with text a sentence that congratulates them on something they accomplished (this should be something they actually did accomplish, find something on their wikipedia page or through google search. Do NOT make something up).

You do not need to provide values for the id, is_read, sent_at or in_reply_to fields, they should be automatically getting set. So your INSERT clause should specify which fields it provides and what their values would be.
6. We now want to "send" to all users a greeting message. Do another INSERT-SELECT query to do the following: Add a new message with text "Welcome to our messaging service" with sender the admin and recipient each user that the admin knows through its acquaintances entries. Start by creating a SELECT query that returns the usernames of all users that are the targets of an acquaintance with source the admin user. Then use that query inside an INSERT. As in 5, your INSERT needs to specify the sender/recipient/message fields, but this time you should **also** specify the is_read field to be set to TRUE.
7. We will now write a "reply" to the message you sent to the FCS (and possibly other messages the FCS has received). Your query should be an INSERT query that does the following:
 - Locates all messages that were sent to the FCS and have not yet been read (via a SELECT).
 - For each such message, creates a "reply" message from the FCS to the sender of that message, with the in_reply_to field set to the id of the received message, and with message text explaining that the FCS apologizes for not being able to reply to all the messages they receive from all their fans.

Follow the INSERT query with an UPDATE query which sets the is_read field for all those messages that the FCS has received to TRUE.

8. Write a SELECT query that lists all the pairs of sender and receiver where there is a message from the sender to the receiver but the sender does not in fact know the receiver according to the acquaintances table. You should see one entry as a response, namely the reply that the FCS sent to you.

You should submit your completed SQL script as an email attachment to me. The name of the file should include your first and last name, in addition to the assignment's number. It should contain no whitespaces.