3344 Database Lab

1. Overview

In this lab, you will:

- Learn (or review) the basics about databases by studying (or skimming) the MySql Tutorial that is linked from this lab in the 3344 labs page.
- Decide what data you will use for your AngularJS project.
- Using MySQL Workbench (an open source GUI for the MySql database management System) design, implement, and populate a database that will support your AngularJS project.
- Write some SQL SELECT Statements.

2. Lab Requirements

A. Create a database that consists of two tables, as follows. **Follow Database table naming conventions:**

- Do not any SQL keywords in your table design (table name, field names) because it will cause unwanted database exceptions later in the semester accessing your database from java code. **Some SQL keywords to avoid: role, user, password, state** (google "SQL Keywords" for complete list)
- Table names should be singular, not plural, e.g. "student" not "students".
- Table names should not include space – you can use underscore if you have a multi-word name, or you can use camel case since the SQL in mySql is case sensitive.

1) A **User Table** with at least these fields:
   - User Id (primary key, auto-increment)
   - User email (they will use as their log on name, must be unique)
   - User Password
   - User Nickname
   - Role Name (e.g., ADMIN or MEMBER).

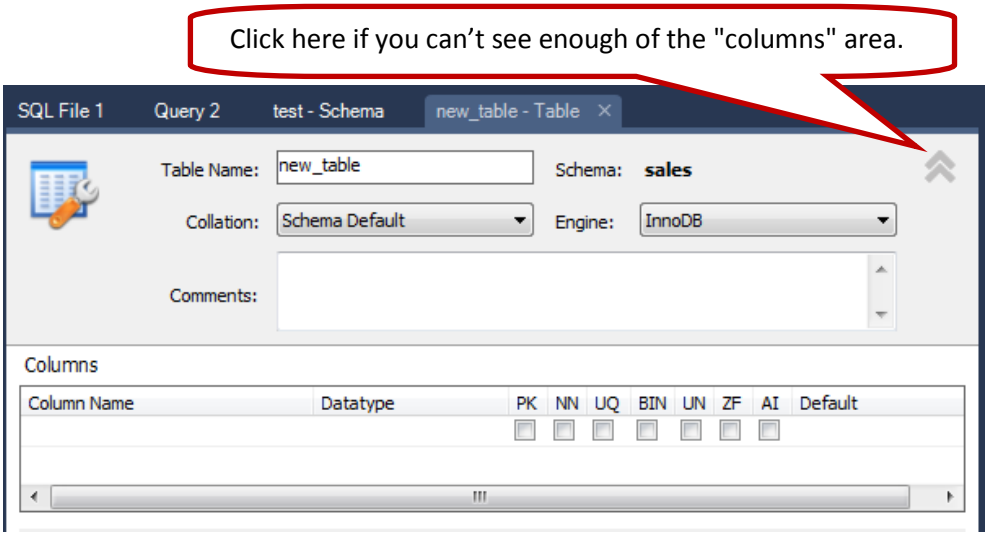
   Populate this table with at least 5 records (realistic looking data). Do NOT name this table "user" (or any upper/lower case version of this SQL keyword). You can call it ***user_t,*** if you like, or something else, but not "user".

2) **An Other Table** that your AngularJS project will access and modify in future labs. Name this table with a self documenting name that represents what you will store (you cannot name it "other").
   - tableName_id (primary key, auto-increment)
   - some sort of descriptor or identifier (must be unique)
   - A field that will hold a long URL to an image.
   - If you wish, you can also have a field that will hold a URL to a website.
   - At least 2 other fields, at least one of which is non-character (either date, integer, or decimal) and nullable.

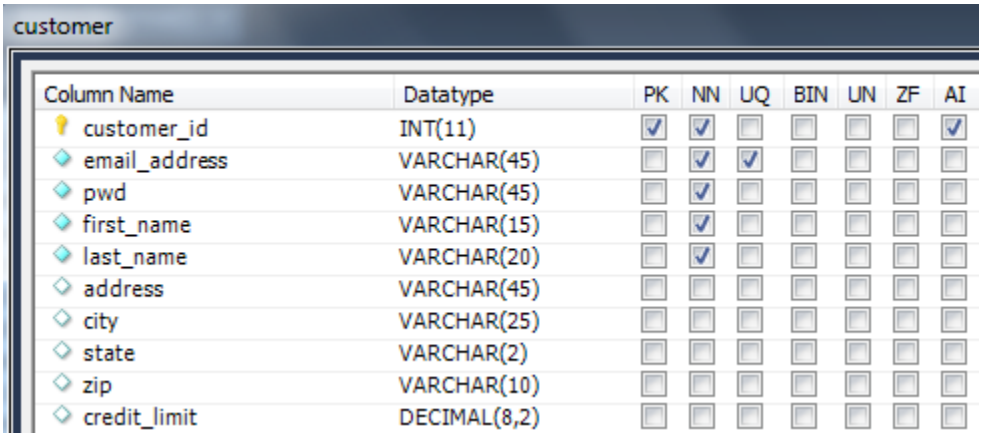
   Populate this table with at least 20 records (realistic looking data). At least one record shall have all fields populated. At least record shall have null in all null-able fields. *Since this table needs to have quite a few records, you may want to search the internet to find some data that you can import. After*

importing, you can modify table design to add other fields and/or you can manually enter some data into these fields for some of the records.

B. Get a screen capture from MySql Workbench of the table design of each of your two tables. To see the table design from MySql Workbench, right click on the table name (left pane) and select "Alter Table".



To get a good (legible) screen capture, you can do an Alt-PrtSc (copies the active window into the clipboard), paste into a simple program like MSPaint, then select just the part you need which should look similar to this:

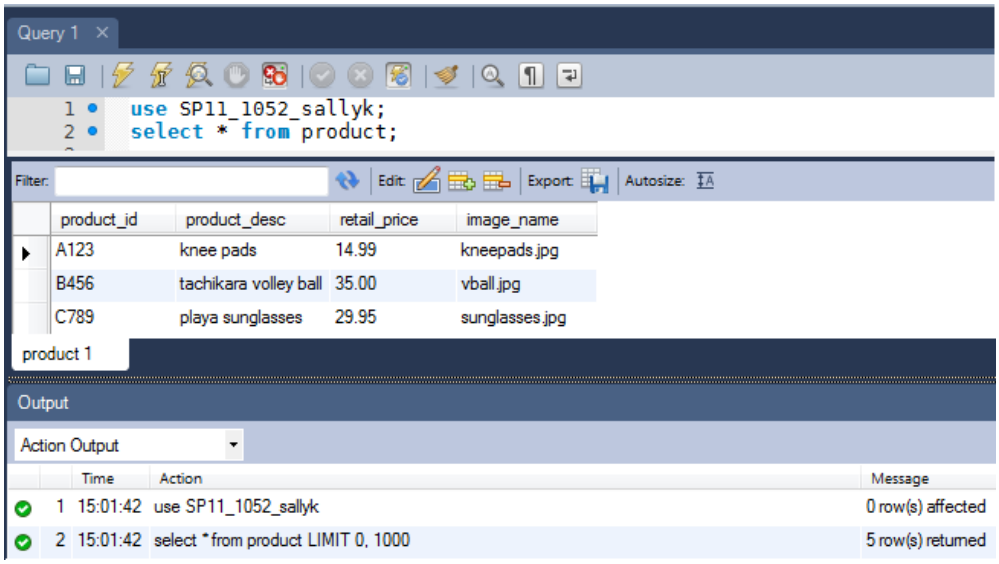


C. Get a screen capture from MySql Workbench of error message that you get when you try to ***insert*** a record into your ***User*** table that has the same descriptor field as some record that already exists in the table. Your web application will get this same database exception error message (if your user tries to do such an insert), so try to get familiar with how the database management system works in this regard.

D. Get a screen capture from MySql Workbench of error message that you get when you try to ***update*** a record from your ***"Other"*** table so that it has the same descriptor field as some record that already exists in the table. Your web application will get this same database exception error message (if your user tries to do such an update).

E. Write and execute the following three **SQL SELECT statements**, then copy/paste a screen capture (sql query area, result set area, plus output area, as shown below) into a document (doc, rtf, or pdf). Get a good (legible) screen capture as mentioned before, by selecting out just the parts that are needed.

NOTE: substantial deductions will be taken if we see syntax errors in the query window or in the output area.



1) **First select statement:** All the fields of all records in your User table, ordered by email address. Select the column names individually (don't use SELECT *). Since users expect the first column to be in order, make the email address be the first column and display the rest of the columns as you think your users (Administrators) would want to see it on the web page you will be creating later in the semester. Even though user passwords should never be displayed to any user (even administrator), we are doing this anyway, for ease of testing and grading.

2) **Second select statement:** All the fields of all records in your Other table, ordered by the unique descriptor field. Select the column names individually (don't use SELECT *). Since users expect the first column to be in order, make the descriptor field be the first column and display the rest of the columns as you think your users would want to see it on the web page you will be creating later in the semester. It is likely that you will not be able to get all 20+ records to show on a single screen capture, so just get the first records on one screen capture and the last records on another screen capture. If you are unable to get all the fields to show in the screen capture, just put the long URLs as the last columns of your select statement and it is OK if we cannot see them.

3) **Third select statement:** The same columns and order as the second select statement, but having an additional condition in the WHERE clause that uses the SQL LIKE keyword and % for wild card match. Choose your WHERE clause so that you get at least several rows in your result set (but not more than can fit in one screen capture). Here is an example using the LIKE SQL keyword that returns all records where user_email starts with "S":

```
SELECT * FROM web_users WHERE user_email LIKE 'S%';
```

3. Submission

Submit the document (doc, rtf, or pdf) to blackboard. The document should be named with your last name in it. This document shall have ***screen captures*** of the following:

- Table design of each of your two tables,
- Two database constraint violation error messages (trying to insert/update and violating uniqueness constraints),
- Each of the 3 SQL select statements listed in the question above.

To your labs.html page, add a blog that (1) describes what you did in this lab and (2) links to your document (the one just above, with your screen captures in it).