1. Overview

In this homework, you will:

- Decide what data you will use for your web application (where your users can log in and see the tabular data from your single database table). Every student will have a `web_user` table and some other table of their own choosing.
- Learn (or review) the basics about databases by studying (or skimming) a MySQL Workbench Tutorial.
  - MySQL is an open source DBMS (database management system) installed on cis-linux2.
  - MySQL Workbench (installed on your PC/MAC) is an open source GUI that lets you connect to the MySQL (running on cis-linux2) — it creates and runs SQL code for you.
- Using MySQL Workbench, design, implement, & populate a database to support your AngularJS project.
- Write some SQL SELECT Statements.
- Create and publish a word document that has screen captures of your database work in this homework, then add a blog (about what you learned in this homework) to your blog page and link to the word document from that blog.

2. Before You Start

In the 3344 Web Page (where you found this document), there are three documents to help you if needed:

1. How to install MySQL Workbench on your home PC/Mac (this will be helpful for you to be able to easily see and modify the data in your database).
2. How to connect to your database (there is a special file created in your home folder under cis-linux2 that holds your auto-generated database password that is different from your NetAccess password).
3. A MySQL Workbench/Database tutorial that tells you how to create tables and modify data.
3. Homework Requirements

A. **Create a document** to hold screen captures as you work through this homework. I will **highlight** all the screen captures that you shall put in the word document.

B. In your database, **create a table named “web_user”** and designed **EXACTLY as shown in the diagram below.** (If you don’t have it exactly the same, sample code will not work for you.) Please note that the relevant column attribute headings are:

- PK means primary key of the table
- NN means “not null” which means the field is required to be provided whenever a record is inserted/updated.
- UQ means unique (cannot have more than one record with same value for this column in this table)
- AI means Auto-increment – the database will assign the next available number when you insert a record (and do not provide a value for this field).
C. In your database, create a second table that is unique to the idea you have for your web application. Name this table with a self-documenting name. Do not name your second table “other” (this is not self-documenting), even though I will refer to this table as “other” in your future homework assignments.

Include at least the following fields in your table:

- xxx_id (primary key, auto-increment), where “xxx” is the table’s name.
- A character 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.

As you design this table:

- 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.

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.
D. **Get a screen capture** from MySQL Workbench of the table design of each of your two tables (and paste it into your document). 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, 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:
E. **Insert web_user Records.** Once you start writing your web application, your server side (java) code will be attempting to insert, update, and delete records. Therefore you need to know enough SQL to accomplish this. The DBMS (database management system) will always protect your data and enforce all the rules you set up when you designed your tables. This type of database protection will throw exceptions in your java code once you start writing server side database access code. So, I’m asking you to use MySQL Workbench to learn about SQL insert/update/delete commands and the database exceptions these operations may cause.

- Be sure to double click on your database on the left pane so that your database is shown in bold. The bolded database name becomes the default database (means you don’t have to prefix every table name with your database name when you type in SQL commands). To help reduce the huge number of databases in the list, you can type in all or part of your database name (e.g., FA18_3344_tua12345) into the filter box under “SCHEMAS”.

- Right click your web_user table (from the database navigator area) and “Select Rows” from your (empty) user table.

- Type in one row of data filling out user_email and user_password but not web_user_id (remember this is an auto-increment type field that the database will supply for you).

- Click the Apply button and check out the INSERT SQL command that MySQL Workbench created for you (to do the insert). It should look something like this (but with your own database name and your own data values):

  ```sql
  INSERT INTO 'FA18_3344_tua12345`.'web_user` (`user_email`, `user_password`) VALUES ('sallyk', 'pw');
  ```

- Click through the process to insert the record and hopefully the record will be successfully inserted (as shown at the bottom pane). The message “1 row(s) affected” means that the insert was successful.
- Type in a second web_user record and, this time as you click through, take special note of the SQL INSERT command that MySQL WB plans to use to insert the record for you.

```
INSERT INTO `sallyk`.`web_user` (`user_email`, `user_password`) VALUES (' ... 
```

- Copy the insert command (from the above screen) into the clipboard, then cancel out the insert operation. Paste the insert command into the SQL query area (above the data grid) and then edit the SQL so that it looks more like the command shown below (delete the unnecessary “back tick” single quotes and remove the database name prefix to your table name). Then Click the lightning bolt icon to run the SQL command.

```
INSERT INTO web_user (user_email, user_password) VALUES ('joe@temple.edu', 'jt');
```

Get a screen capture that shows a successful insert – the output area should say “1 row(s) affected”. Your screen capture should include the SQL INSERT command (at the top), the data grid (in the middle) and the output confirmation message (at the bottom).

- Try to execute the same INSERT statement again and you should get an error message because this insert would violate the email address uniqueness constraint that you added to your user table. Get a screen capture that shows the database “throwing an exception” and not allowing the insert to occur. This screen capture just needs the bottom part (error message).

Your server side java code might experience this type of database exception and it has to know how to communicate the problem back to the client side so that your JavaScript code can tell the user what the issue is.
F. **Update user records.** Using the grid, update a user record. Do the same trick (learning SQL commands from MySQL Workbench) – copy the UPDATE command before clicking “Apply” the second time. Paste the UPDATE command and edit it (remove back ticks and remove the database prefix before the table name). Run this update command by clicking on the lightning bolt. Get a screen capture showing that you successfully ran your own UPDATE command on your user table.

G. **Delete user records.** Using the grid, click to the left of a data row, then right click and select “delete”. Do the same trick (learning SQL commands from MySQL Workbench) – copy the DELETE command before clicking “Apply” the second time. Paste the DELETE command and edit it (remove back ticks and remove the database prefix before the table name). Run this update command by clicking on the lightning bolt. Get a screen capture showing that you successfully ran your own DELETE command on your user table.

H. Perform additional data entry so that you have 5-7 realistic looking web_user records.

I. **Insert “other” records.**

- Populate your “other” table with around 15 records of realistic looking data. Make sure that at least one record has all fields filled out and at least one record has null in all nullable fields.
- Note: In your next homework, you’ll be writing java code to get the data from the database and display it. So, the data in your image URL field will need to actually point to valid images in order for you to be able to turn these URLs into actual pictures on your web page.
J. Write and execute the following three SQL SELECT statements, then get a screen capture of each and paste this screen capture into your document. Each screen capture shall include: SQL Query area, Result Set area, plus Output area, as shown below. To get a good (legible) screen capture, size the areas of MySQL WorkBench as small as possible showing only the data you need to show, then Alt-PrtSc as mentioned before, and select out just the part you need to show.

- **First select statement:** All the fields of all records in your web_user table, ordered by email address. Select the column names individually (don’t use SELECT *) beginning with the email address which is the “ORDER BY” column.

- **Second select statement:** All the fields of records in your Other table, ordered by the unique descriptor field. Select the column names individually (don’t use SELECT *) beginning with the descriptor field which is the ORDER BY column and followed by the other columns in the order you think your users would want to see the data on a web page. Add a WHERE clause so that only about 5-10 records are selected.

- **Third select statement:** Starting with the same SQL from your second select statement, replace the WHERE clause so that it 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%';
```

**NOTE:** substantial deductions will be taken if we see syntax errors in the query window or in the output area.
4. Recap (word document contents)

The document with your screen captures shall be named with your last name in it and it shall include the following:

- Table Designs (from MySQLWorkBench):
  - Screen capture your user table design.
  - Screen capture of your other table design.
- INSERT/UPDATE/DELETE user:
  - Screen capture of successful insert (after you typed in the insert command to the Query Area).
  - Screen capture of insert that failed due to uniqueness constraint (with you typing the SQL into the Query Area again).
  - Screen capture of successful update (you typing SQL into Query Area)
  - Screen capture of successful delete (you typing SQL into Query Area)
- SQL select statements: Screen capture of each of the 3 SQL select statements listed on the previous page.

5. Publishing

To your blog page, add a blog that (1) describes what you did in this homework and (2) links to the document described just above. You link to the document just like you would like to a html page, for example:

Click `<a href="smithDataBase.doc">here</a>` to see my database work.

6. Submission

As always, publish your work and test that your blog is there and links correctly to the word doc. Then submit a zip file of your whole project into blackboard. Start getting used to submitting the WHOLE project (not just the “web pages” folder) since beginning next week, the whole project becomes part of the necessary source code.