3344 Database Homework

1. Overview

In this homework, you will:

- Decide what data you will use for your web application. 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 data into your two table database.
- 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

- After connecting to MySQL through MySQL Workbench, **set your database schema to be the default schema** (and then you should see your database schema shown in bold). To do this, double click on your database schema or right click your database schema and select “set as default schema”. So that you don’t have to scroll through zillions of databases, you can type in all or part of your database name (e.g., SP19_3344_tua12345) into the filter box under “SCHEMAS”.

![MySQL Workbench](image)

After setting your database schema as the default schema, it should show up bolded in the Navigator pane.

To create a new table, open up your database schema (click on triangle) then right click Tables and select “Create Table”.

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).

![Table Screenshot](image)
C. Insert web_user Records.

- 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, this time filling in all the columns. MySQL wants you to type in dates in this format: YYYY-MM-DD. Note: put in any values you want – you don’t have to use the data values I show here:

<table>
<thead>
<tr>
<th>web_user_id</th>
<th>user_email</th>
<th>user_password</th>
<th>birthday</th>
<th>membership_fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="mailto:ioe.blow@temole.edu">ioe.blow@temole.edu</a></td>
<td>dw</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>2</td>
<td>test</td>
<td>test</td>
<td>1998-02-03</td>
<td>123.08</td>
</tr>
</tbody>
</table>

- Type in more records until you have 5-7 realistic looking records in your web_user table.
D. In your database, **create a second table** that is designed to hold data that makes sense for the topic you chose 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 (specify this as a unique type field)
- A long varchar field to hold a 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.

E. **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: So that your web application can actually show images, make sure that the image URLs are valid and point to images out in the internet (use fully qualified URLs like http://...)


F. 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.

- **Third select statement:** Add a WHERE clause to your second select statement so that not all of your other records are selected.
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.
- SQL select statements: Screen capture of each of the 3 SQL select statements listed on the previous page.

Note: to get a screen capture, press Alt-PrtSc (hold down the Alt key while pressing the PrtSc key). This copies the active window into the clipboard. Then paste the screen capture into a program such as MSPaint. From MSPaint, select / copy just the part you need (e.g., the table design) then paste into your homework document. If you try to paste the whole window directly into the HW document, usually the part that we need to see is too small to see.

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.

NOTE: This is a VERY EASY Homework – spend extra time working on your Tutorial...