PHPM 631 Lab 5 & Assignment 5: Writing SQL Queries (w3schools.com)
Lab 5 - Due date: Submit on E-Campus by 11:59pm Monday 2/27 & 3/6
Assignment 5 - Due date: Submit on E-Campus by 11:59pm Sunday 3/12

Submission. Submit on E-Campus. See Requirements Section Below for details

- Lab 5, Week 1 (2/27): w3schools.com
  - Upload your tutorial progress report
- Lab 5, Week 2 (3/6): Writing SQL Queries
  - Upload your own SQL Queries to answer the questions given (May work in groups)
- Assignment 5 (3/12)
  - Week1 (3/5) - w3schools.com: Upload the results of your SQL Quiz
  - Week2 (3/12) - Upload your SQL queries code (May work in groups)

Late Assignments: Due to the scheduled midterm, no late assignments are allowed for this assignment. Submit whatever you have done by the due date.

Plagiarism: If you consult any outside sources when doing your work, you are expected to further document these sources. Give credit where credit is due. Plagiarism will not be tolerated.

Guideline for lab grading

You must submit whatever you did during in class lab on blackboard by the end of day of class. These will be include in the in class participation evaluation which is part of the grade for the last assignment.

Guideline for assignment grading (8%)

- 70% (70 points): Bad (√ --) Did NOT follow all instructions
- 80% (80 points): Reasonable (√ -) Followed all instructions
- 90% (90 points): Good (√) Followed all instructions, and did good work
- 100% (100 points): Great (√ +) Followed all instructions and did great work

Objective

By the end of this assignment, you should be able to

- Write SQL queries to answer specific questions about the information in the database

Assignment 5 & Lab5: Writing SQL Queries

It's now time to look data

There are two parts to this lab and assignment. Learning through w3schools.com and then applying what you learned to your own SQL queries.

w3schools.com (Week 1)

Work through the following "SQL" tutorial on w3schools.com (http://www.w3schools.com/sql/default.asp).

Recommended Action Plan

1. Open your browser
2. search for "w3school"
3. click on "SQL" tutorial (on top)
4. Work through the following sections under SQL Tutorial (just read the page and follow the instructions. Click on "try it yourself" and try different queries so you understand how to write queries)
   a. SQL HOME
   b. SQL Intro
   c. SQL Syntax
   d. SQL Select
   e. SQL Distinct
   f. SQL Where
   g. SQL And & Or

    h. SQL Order By
    i. SQL Like
    j. SQL Wildcards
    k. SQL Joins
    l. [OPTIONAL] SQL Inner Join
    m. [OPTIONAL] SQL Left Join
    n. [OPTIONAL] SQL Right Join
    o. [OPTIONAL]SQL Full Join
    p. SQL View

5. Submit progress report for *lab 5 Week 1*. See “Required Submissions” for details.
6. Work through the following sections under SQL Functions (just read the page and follow the instructions)
   - a. SQL Functions
   - b. SQL Avg()
   - c. SQL Count()
   - d. SQL First()
   - e. SQL Last()
   - f. SQL Max()
   - g. SQL Min()
   - h. SQL Sum()
   - i. SQL Group By
   - j. SQL Having
   - k. SQL Ucase()
   - l. SQL Lcase()
   - m. SQL Mid()
   - n. SQL Len()
   - o. SQL Round()
   - p. SQL Now()
7. Submit code for Part 1 for *assignment 5*. See “Required Submissions” for details.

Required Submissions

1. Whenever you are done for the class (lab 5, week 1): write a short progress report in a text file noting how far you got in the tutorial. Just a simple list of sections you have completed will suffice.
2. When you are done going through all the required sections in the tutorial: Go back to the SQL Home page and take the SQL Quiz Test (http://www.w3schools.com/sql/sql_quiz.asp). When you are done, “Check your answers” and print to pdf. Submit this as assignment 5 week 1. Do the best you can, but don’t worry if your score is not high. This is a LEARNING activity and NOT a grading activity. Everyone who submitted evidence of completing the quiz will get the same grade. However, I encourage you to talk to your classmates and TA about ones you got wrong and learn the right answer. As if you have not properly learned this material, than you will inevitably not do well on the graded assignment for week 2 and the final exam.
3. NOTE: There are two submissions. One for Lab 5 week 1 (when you are done on Monday) and another one for Assignment 5 week 1 (when you are done with everything).

Writing SQL Queries (Week 2)

Now it is time for you to write your own queries. If you have learned properly last week, this week should be easy. If not, you should take this week to learn what you missed last week.

You are encouraged to work in a team of 2 for this assignment. If you do, both of you will receive the same grade for this part of the assignment. It is IMPORTANT that BOTH of you submit on blackboard. Each of you are REQUIRED to type your own SQL query without copy/pasting. BUT at the very top, clearly label that you worked as a team and the name of the team members. You may choose to work along, but you will be expected to do the same amount of work as a two-person team.

YOU WILL HAVE TO WRITE SQL QUERIES FOR THE MIDTERM.

In this assignment, you will be using SQLite3 (a variation of SQL) to write queries to answer questions given below using a database I provide. You have THREE options of how to use SQLite. Below, I give you instructions for using a cloud based system, which sometimes can be slow. If you prefer to install a small software on your own computer, at the end of the assignment are instructions for how to install on a windows based machine and a Mac.

How will you use SQLite3 (Recommended. See last section for another option)

1. Google "coding ground", and select SQLite (http://www.tutorialspoint.com/execute_sql_online.php)
2. On the top right, you should see a menu
3. In the upper left region, you will see a folder named "root" (this is your home directory). Right click on "root", and select "upload file" and upload the "kumdb.sql" database (get this from the class website)
4. Make the bottom window ("Terminal") as big as you can
5. Then click on the "Terminal" window (you should see a cursor blinking) so you can type into it
6. Your prompt should say something like **`sh-4.2#`**. This indicates you are at a shell (OS)
7. Type
   ```
   sh-4.3$ sqlite3 kumdb.sdb < kumdb.sql (enter)
   sh-4.3$ sqlite3 kumdb.sdb (enter)
   ```

   kumdb.sql is available on the class website. You should see something like the following on the screen.
   ```
   SQLite version 3.8.7 2014-10-17 11:24:1
   Enter ".help" for usage hints.
   sqlite>
   ```
8. Note now the prompt has changed to **`sqlite>`** This indicates you are now using the SQLite software, and no longer at the shell. You have loaded the kum.sdb database into the SQLite software.
9. At the **`sqlite>`** prompt you can type either SQL statements (followed by semicolumn), or special commands that SQLite can understand (these all start with a dot in the front).
10. For example, when you are done, type **`.quit`** (don't forget the dot in the front). This will quit SQLite and return back to the shell.
11. Or **`.help`** will print the SQLite commands on the screen. But if your window is small, it will scroll by. So make your window big. Our class website has the list of commands you need to know.
12. You can save the project to various locations as well as download project to local machine. But personally, I find it is better to just use a "fresh" shell and repeat the following steps each time because it is simple. But, it might help for you to keep record of your SQL queries as you work on them in a separate text document so you don't lose them.
    - upload kumdb.sql to new session
    - run it to create kumdb.sdb: **`sqlite3 kumdb.sdb < kumdb.sql`** `(enter)`
    - start SQLite loading in kumdb.sdb: **`sqlite3 kumdb.sdb`**
    - [OPTIONAL] **`.header on`**
    - [OPTIONAL] **`.mode column`**
13. TIP: in the green command line, you can use the arrow keys to *recover* previous commands you typed in. This is called the history of your commands. You can reduce typing by traversing the command history. Try using the arrow key to see what it does. When you find a command you want to rerun, just hit **`Enter`**.

14. If you see the following message, **`Disconnected! Trying to reconnect with the server...`**, just reload the page

Recommended Action Plan

1. First, do step 12 above
2. See what tables you have in the database: **`.tables`**
3. Look at all the tables, one table at a time, using SQL queries. Remember your semicolon at the end. (HINT: **`select * from table_name;`**)
4. Now you should know what is in the database.
5. Answer the questions in the Required Submission section using SQL queries. Remember to note the answer AND the SQL query that you used to answer the question as you go in a document for submission.
6. When you are done for the session type **`.quit`** and quite SQLite.
7. Close the browser. Repeat 1-6 until you are done with your homework.

Required Submissions

1. Lab 5, Week 2.
   a. Confirm you were able to use code ground to write a simple SQL query by answering the following questions: What tables do you have in kum.sdb? How many doctors are at this clinic? What SQL query did you use to find this out?
2. Assignment 5 Part 2: Answer the following questions AND submit your SQL queries that were used to find the answer. That is in a separate text file, copy and paste on final SQL that answers the question, then answer the question and move to the next question. **One text file of all your SQL queries and answers to questions are what you submit.** NOTHING else is submitted. You must use SQL queries to answer all questions. If you can not write your SQL queries, still submit the answers to the questions for partial credit.

   At the TOP: List group members, if any, you worked with. But remember, you MUST do your own typing.

CAUTION. If you changed the database by accident, stop and exit from everything. And start over so that you are using the correct database. Remember to delete kum.sdb and upload again. This is a small database, so you should be able to answer the questions manually without writing the SQL queries. You might want to think of this assignment as writing the correct SQL queries to get the same results as your manual answer, noting that you could have made a mistake in your manual answer. So if your SQL query result is not matching your manual answer, think about why and figure out which was wrong. Note that conversely, even if you get the exact same answer as your manual answer the SQL query might not be fully correct. That is, given the particular DB, you got the correct answer, but for a different DB it might not be correct.

Q1: Write a query that displays an alphabetical list (by last name) of all patients.

Q2: Write a query that displays the names of all patients who are Hispanic. Save this as a view to use later.

Q3: Using your saved query (i.e., view) from Q2, display the names of all patients who are Hispanic ordered by last name

Q4: Write queries to answer each of the following questions (you can write as many queries you need to answer the questions):

1) How many visits did Lila Autry have? Who saw her on each visit? (*Hint: which columns, from which tables do you need to answer this question?)*
2) Who is her primary care doctor (the variable **`primary_dr`** holds this information) ?
3) What were her diagnosis (all of them)? (*Hint: You will have to use the lookup table to find the meaning of the diagnosis code)*
4) Were any medications prescribed? If so, how many and what?

Q5: Write a query that displays patients who visited in the second quarter of 2012 together with the number of visits that quarter. Order the output by the number of visits (patients with the largest number of visits should be displayed first). Each patient name should appear once.

Hint: Join tables "patients" and "visits" and use Group By and Count. You'll also need to use a Where for the date range and Sort on (Order by) the Count.

Q6: Write a query that displays the **`billed, covered,`** total paid (=**`copay+pat_pd+insur_pd`**), and unpaid amount (=**`covered`**-total paid) for each patient. Save this as a permeant view so that you can use this again in the next task.

Hint: You'll need to calculate new columns for the total paid amount and the unpaid amount. You'll also have to Group By and Sum.

Q7: Using the view from Q6, write a query that displays which patients have not paid their billed amount yet, and what is the outstanding bill amount? Order the output by outstanding bill amount with large amount on top.

Hint: Join tables "patients" with the view and sort. You'll also need to use a Where to find the patients with outstanding bills. Try using greater than 1 cent.

Using SQLite3 with a Portableapp (to install on any computer including lab machines)

If you do not want to use the cloud tool or it is not working, here is another option. Portableapps are software that do not need to have administrative rights to install and use on a computer. For the most part, you download onto any folder on the computer and install into any folder, then just run the software from there. You can even "carry" your software around on a USB drive (or on your dropbox) and run the software on any computer directly from those locations. If you feel that the software is slow, you might want to make a local copy of the full folder and run it from the computer.

The following only run on windows.

1. Go to: portableapps.com/apps/development/sqlite_database_browser_portable
2. Click on green icon to Download
3. Install into a local folder (e.g. desktop or usb drive). You can copy and paste the folder once installed
4. Run SQliteDatabaseBrowser (extension is .exe)
5. Once launched click on "Open Database" and upload the "kum.sdb" database (from the class website).
6. To run sql queries click on "Execute SQL". Write your queries and click on forward button (▶) to run.
7. You can also run using F5 or Ctrl+enter.
8. Copy and paste your final SQL queries into a separate text file. Collect all SQL queries for your questions into one text file and submit with the answers to questions.
9. You can save the project to various locations as well as download project to local machine. But you are not making any changes to the DB so there is no need to save anything. Personally, I find it is better to just use a "fresh" shell and repeat the following steps each time because it is simple. But, it might help for you to keep record of your SQL queries as you work on them in a separate text document so you don't lose them. **Remember to upload kum.sdb to any new session.**
10. A text editor portableapp: http://portableapps.com/apps/development/notepadpp_portable

For Mac users, please follow the steps below to download the software:

1. Go to: sqlitebrowser.org/
2. Please download the 'SQLCipher Mac.dmg' by clicking the fifth blue icon on the right side.
3. Find the downloaded software in the 'Downloads' and move to a folder you prefer to use.
4. Double clicks or click 'Open' to run the 'sqlitebrowser_sqlcipher-3.8.0v5.dmg'. There may be a pop-up window asking you whether you sure you want to open it. Please click 'Open', and then you can play with SQL.